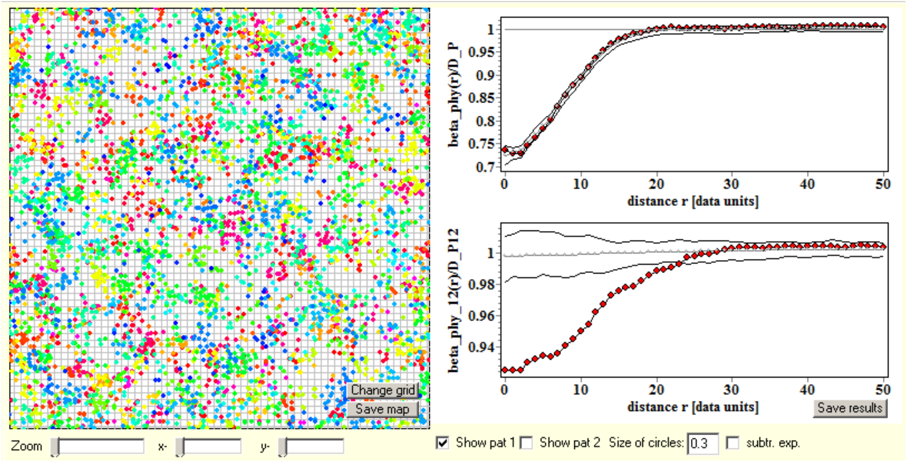
User Manual for the *Programita* software



☑ Show pat 1 □ Show pat 2 Size of circles: 0.3 □ subtr. exp.

Version
November 2018

written by
Thorsten Wiegand
Department of Ecological Modelling,
Helmholtz Centre for Environmental Research - UFZ,
Permoserstr. 15,
04318 Leipzig, Germany,

Tel.: (**49) 341 235 1714
thorsten.wiegand@ufz.de

Contents

1 General information

Programita is a comprehensive software package for conducting spatial point pattern analysis in ecology. I tailored *Programita* to accommodate the needs of "real world" applications in ecology and developed the different modules in response to my own research questions and to requests of colleagues and students who have approached me with their specific research problems in mind. More than ten years after its launch in the 2004 Oikos Mini review (Wiegand and Moloney 2004), *Programita* has considerably grown and now contains a variety of statistical methods for most of the **point pattern data types** which are relevant in ecological applications, including

- **univariate patterns** (i.e., one type of points)
- **bivariate patterns** (i.e., two types of points such as two species of trees)
- **multivariate patterns** (i.e., several types of points such as a forest tree community)
- **multivariate patterns** with a matrix of pairwise (e.g., functional or phylogenetic) dissimilarities between types of points
- **qualitatively marked patterns** (i.e., one type of point with a qualitative mark such as surviving vs. dead)
- **quantitatively marked patterns** (uni- or bivariate pattern augmented with quantitative marks such as size)
- **objects with finite size and real shape** (cases for which the point approximation does not hold)

Programita offers for each of these data types the most appropriate **summary functions**:

- **univariate patterns** [pair correlation function $g(r)$, L-function $L(r)$, the *K2* function $K2(r)$, the distance distribution functions $D^k(r)$ of the kth nearest neighbor, the spherical contact distribution $H_s(r)$, the mean distance to the kth neighbor $nn(k)$, and inhomogeneous g- and L-functions]
- **bivariate patterns** [pair correlation function $g_{12}(r)$, L-function $L_{12}(r)$, the *K2* function $K2_{12}(r)$, the distance distribution functions $D^k{}_{12}(r)$ to the kth nearest neighbor, the mean distance to the kth neighbor $nn_{12}(k)$, and inhomogeneous g- and L-functions]
- **multivariate patterns** [e.g., spatially explicit Simpson index $\beta(r)$, individual species-area relationship $ISAR_f(r)$, distance decay of similarity $F(r)$]
- **multivariate patterns with dissimilarity matrix** [e.g., phylogenetic Simpson index $\beta_{\mathrm{phy}}(r)$, phylogenetic mark correlation function $k_d(r)$, phylogenetic ISAR function $PISAR_f(r)$, rISAR function]
- **qualitatively marked patterns** [mark connection functions $p_{ij}(r)$ and various test functions based on pair correlation or K-functions]
- **quantitatively marked patterns** [various normalized and non-normalized mark correlation functions $k_t(r)$]
- **objects with finite size and real shape** [$g(r)$, $L(r)$, $g_{12}(r)$, $L_{12}(r)$]

and the most important **null models** for ecological applications, which often allow for consideration of a spatially varying intensity function $\lambda(\mathbf{x})$, and a variety of **point process models** describing clustering and regularity that are relevant for ecological applications.

Programita allows you to conduct Monte Carlos simulations of null model point processes, fit cluster point processes to the data, and to generate stochastic realizations of the point processes to determine pointwise and global **simulation envelopes** and to conduct several **Goodness-of-Fit (GoF)** tests.

1.1.1 Terms of use

I am not in the commercial software business, but recognize the need for a tool like *Programita* for scientists to assist in research on spatial point pattern analysis. I produced the *Programita* software to foster the analysis of point patterns in ecology and to provide ecologists with a tool that contains null models and procedures not supported by most statistical packages, but which are essential for a thorough analysis of point-patterns. I have done my best to provide in the documentation complete, step-by-step instructions for the variety of analysis you can conduct with *Programita*, but the user is responsible for acquiring the necessary background knowledge to appropriately use the software.

The *Programita* software may be downloaded and used free of charge for purposes of scientific research and teaching. However, please do not distribute the link, *Programita* or the manual. Any commercial application of *Programita* requires my previous permission by the author. *Programita* is provided “as is” without warranty of any kind. In no event will the authors be liable for any damages, including lost profits, lost savings, or other incidental or consequential damages arising from the use of or the inability to use this software.

Publications using *Programita* must acknowledge use of *Programita* and include the following citations:

Wiegand T., and K. A. Moloney 2004. Rings, circles and null-models for point pattern analysis in ecology. *Oikos* 104: 209-229.

Wiegand T., and K. A. Moloney 2014. A handbook of spatial point pattern analysis in ecology. Chapman and Hall/CRC press, Boca Raton, FL.

if analysis of objects with finite size and real shape is used add:

Wiegand, T., Kissling, W.D., Cipriotti, P.A., and Aguiar, M.R. 2006. Extending point pattern analysis to objects of finite size and irregular shape. *Journal of Ecology* 94: 825-837

if cluster point processes are used add:

Wiegand, T, A. Huth., and I. Martínez. 2009. Recruitment in tropical tree species: revealing complex spatial patterns. *The American Naturalist* 174: E106 - E140

if random labeling analysis is used add:

Jacquemyn, H., P. Endels, O. Honnay, and T. Wiegand. 2010. Evaluating management interventions in small populations of a perennial herb *Primula vulgaris* using spatio-temporal analyses of point patterns. *Journal of Applied Ecology* 47: 431–440

for analysis of multivariate patterns add:

Shen, G., T. Wiegand, X. Mi, and F. He in 2014. Quantifying spatial phylogenetic structures of fully mapped plant communities. *Methods in Ecology and Evolution* 4: 1132-1141

Wiegand, T., M. Uriarte, N.J.B. Kraft, G. Shen, X. Wang, and F. He. 2017. Spatially explicit metrics of species diversity, functional diversity, and phylogenetic diversity: insights into plant community assembly processes. *Annual Review of Ecology, Evolution, and Systematics* 48:329–351

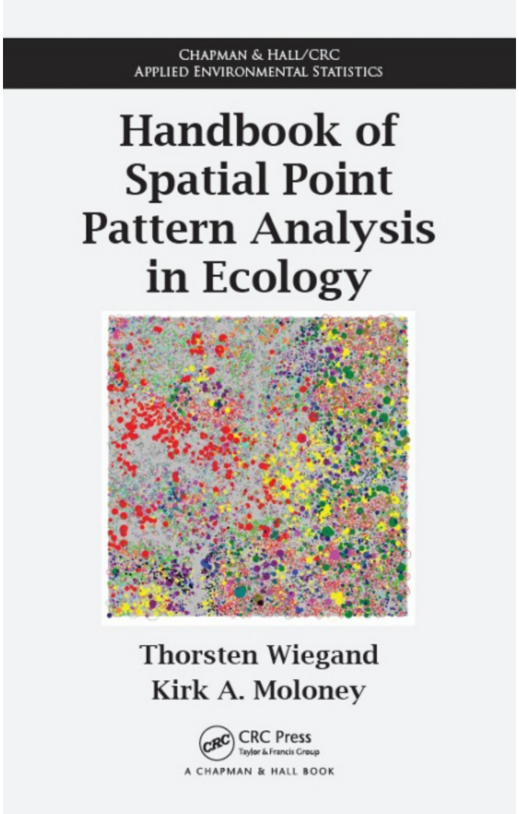
If global simulation envelopes are used add:

Wiegand, T., P. Grabarnik, and D. Stoyan. 2016. Envelope tests for spatial point patterns with and without simulation. Ecosphere 7(6):e01365

1.1.2 Handbook of Spatial Point Pattern Analysis in Ecology

Understand How to Analyze and Interpret Information in Ecological Point Patterns

The methods underlying *Programita* and many examples executed with *Programita* can be found in our recent book published by Chapman and Hall. This manual will therefore not contain detailed explanations of the methods, but I will refer instead to the respective sections in the book. **I warmly recommend you to buy the book to fully benefit from the possibilities offered by *Programita*.**


CHAPMAN & HALL/CRC
APPLIED ENVIRONMENTAL STATISTICS

Handbook of Spatial Point Pattern Analysis in Ecology

Thorsten Wiegand
Kirk A. Moloney

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Although a broad array of statistical methods for analyzing spatial point patterns have been available for several decades, they haven’t been extensively applied in an ecological context. Addressing this gap, Handbook of Spatial Point Pattern Analysis in Ecology shows how the techniques of point pattern analysis are useful for tackling ecological problems. Within an ecological framework, the book guides readers through a variety of methods for different data types and aids in the interpretation of the results obtained by point pattern analysis.

Ideal for empirical ecologists who want to avoid advanced theoretical literature, the book covers statistical techniques for analyzing and interpreting the information contained in ecological patterns. It presents methods used to extract information hidden in spatial point pattern data that may point to the underlying processes. The authors focus on point processes and null models that have proven their immediate utility for broad ecological applications, such as cluster processes.

Along with the techniques, the handbook provides a comprehensive selection of real-world examples. Most of the examples are analyzed using *Programita*, a continuously updated software package based on the authors’ many years of teaching and collaborative research in ecological point pattern analysis. *Programita* is tailored to meet the needs of real-world applications in ecology. The software and a manual are available online.

Features

- Focuses on the application of spatial point pattern analysis in an ecological context
- Helps ecologists unfamiliar with advanced statistics select the proper analysis method
- Emphasizes the formulation of appropriate null models and point processes for describing the features of point patterns and testing ecological hypotheses of spatial dependence
- Provides the *Programita* software package on the first author’s website, enabling readers to perform analyses with their own point pattern data
- Includes a collection of real-world examples
- Offers suggestions on how to use the book for teaching graduate students

1.2 Before starting *Programita*

1.2.1 Hardware requirements

Programita is a free unsupported software, developed in Borland Delphi under a Windows 7 environment. *Programita* is also executable under older Windows versions such as Windows XP and can be used under a MacOs with help of programs like WinneBottler (http://winebottler.kronenberg.org) and Xquartz (http://xquartz.macosforge.org/landing/).

1.2.2 Installation

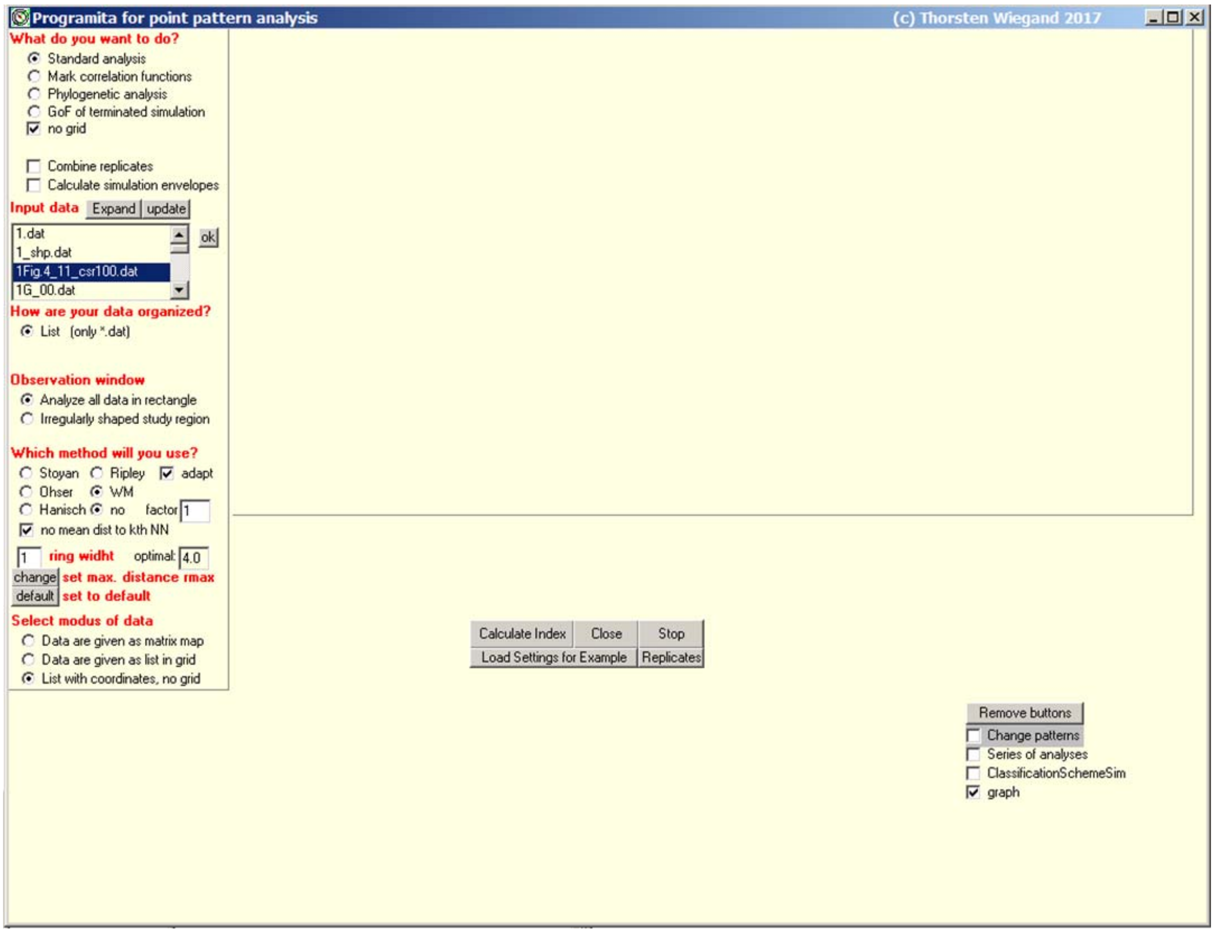
There is no setup procedure; installation of *Programita* requires only the extraction of all files from the zip file ProgramitaOctubre2018.exe. Place the files into a directory of your choice; extracting the zip file will place all files into the sub-directory *Programita*. Note that you must place *Programita* into the same directory as the data input files; for simplicity *Programita* does not use a path variable. However, since *Programita* occupies little space you can place into each folder with data files for specific analysis a copy of *Programita*.

The zip-file contains the following files and file types:

ProgramitaNoviembre2018.exe	• the executable of *Programita,* version November2018
*.dat files	• data files for uni-, bivariate and qualitatively marked analyses and temporary files
*.mcf files	• data files for mark correlation analysis
*.phy files	• data files for multivariate analysis
*.asc files	• data file in ArcView raster format for grid-based analyses of objects with finite size and real shape
*.irr files	• text file with coordinates to encircle an irregularly shaped observation window
*.txt files	• data files with the dissimilarity matrix and species list for multivariate analysis
*.spl	• text files with list of species to be analyzed in ISAR analysis
*.res files	• files to be used to load the settings of an analysis and to redo an analysis. Contain the results of an analysis and all its settings.
*.rep files	• data files to show results of previous analyses and for combining replicates
*.int files	• plug-in files with the intensity function $\lambda(\mathbf{x})$, for example to be used in the heterogeneous Poisson process
*.env files	• temporary files containing the observed summary functions and that of the simulations of the null model; required for GoF and envelope tests and result graphs in *Programita*

1.2.3 Screen size

Programita occupies a screen of 1024 × 784 pixels and after executing *Programita* the screen shown below should appear. However, sometimes buttons or windows within *Programita* are truncated and the text does not fit. To avoid this problem check the default letter size in the settings of your computer. Your computer may scale the letters but not the window sizes and as a consequence, the windows appear too small.


Programita for point pattern analysis
(c) Thorsten Wiegand 2017
Expand
update
ok
change
default
Calculate Index
Close
Stop
Load Settings for Example
Replicates
Remove buttons

What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

□ Combine replicates
□ Calculate simulation envelopes

Input data
1.dat
1_shp.dat
1Fig.4_11_csr100.dat
1G_00.dat

How are your data organized?
● List (only *.dat)

Observation window
● Analyze all data in rectangle
○ Irregularly shaped study region

Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1
☑ no mean dist to kth NN

1 ring widht optimal: 4.0
set max. distance rmax
set to default

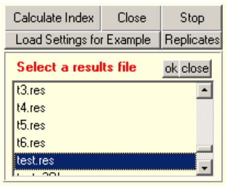
Select modus of data
○ Data are given as matrix map
○ Data are given as list in grid
● List with coordinates, no grid

□ Change patterns
□ Series of analyses
□ ClassificationSchemeSim
☑ graph

2 Features of *Programita* common to all analysis modes

2.1 Load a settings file to redo an analysis

There is a convenient way to quickly start with *Programita* and to learn the settings. You can read a file (a *.res-file) that contains all setting of a previous analysis and redo this analysis. To do this, click button "**Load settings for Example**", select a results file (in the example "test.res"), and the small "**ok**" button. *Programita* now reads all settings from the file test.res and if you click "**Calculate Index**" *Programita* repeats the analysis. However, this works only if all data files are in the same folder.

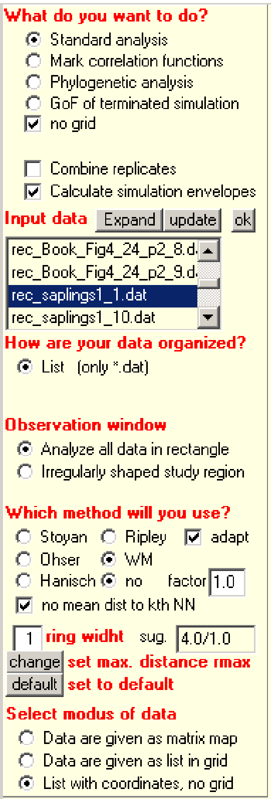


2.2 Overview over menus

Programita allows you to conduct analyses of a variety of data types. Basically, you can use:

1. the **Standard analysis** mode for uni-, bivariate and qualitatively marked analyses
2. the analysis mode for **Mark correlation functions**
3. the analysis mode for multivariate analysis using a dissimilarity matrix (**Phylogenetic analysis**)
4. the grid-based standard analysis mode (box "no grid" not checked)
5. the analysis model for object of finite size and real shape ("no grid" not checked)

You can select the data type in the window What do you want to do?



What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

☐ Combine replicates
☑ Calculate simulation envelopes

rec_Book_Fig4_24_p2_8.d
rec_Book_Fig4_24_p2_9.d
rec_saplings1_1.dat
rec_saplings1_10.dat

How are your data organized?
● List (only *.dat)

Observation window
● Analyze all data in rectangle
○ Irregularly shaped study region

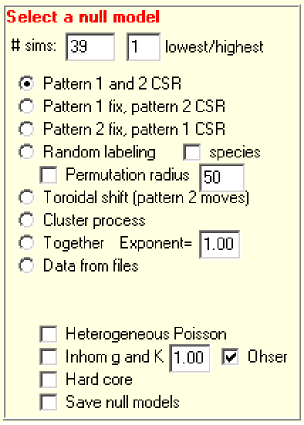
Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1.0
☑ no mean dist to kth NN

1 ring widht sug. 4.0/1.0

Select modus of data
○ Data are given as matrix map
○ Data are given as list in grid
● List with coordinates, no grid

The settings are governed by two different sets of menus. First, one menu bar (at the left side of the screen) allows you to select the type of analysis and the estimators. For example, in the menu for the **standard analysis mode** shown on the left you can:

- select "**Standard analysis**" in What do you want to do? to reach the standard analysis mode. With "**no grid**" disabled you reach the grid-based standard analysis mode
- select the input data file (Input data)
- provide information on the organization of your data (How are your data organized?) and (Select modus of data)
- analyze observation windows with rectangular or irregular shape (Observation window)
- select the estimators of the summary functions (Which method will you use?)
- define a ring width for estimation of the pair correlation function and a maximal distance *r* of analysis. *Programita* suggests an initial ring width (for pattern 1/pattern 2) based on equation 4.3.43 in Illian et al. (2008): $dw = 0.2/\lambda^{0.5}$ for almost random patterns. However, for regular or hyperdispersed patterns the ring width can be smaller.



Select a null model

sims: 39 1 lowest/highest

● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

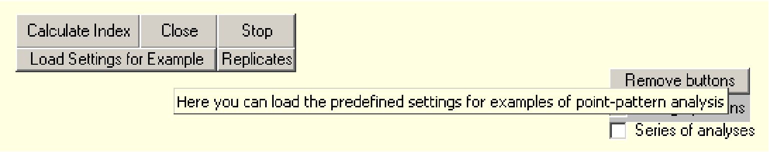
□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

Second, if you check the checkbox "**Calculate simulation envelopes**" located above "Input data" a menu window that appears on the bottom right that allows you to specify the settings of the null model of the standard analysis.

The radio buttons are the different null models available for this data type (e.g., "**Pattern 1 and 2 random**" uses the CSR null model for pattern 1 and pattern 2), and the checkboxes specify settings of the null model selected (e.g., with "**heterogeneous Poisson**" you can select an intensity file for the heterogeneous Poisson process).
On the top you can select the number of simulations of the null model **(# simulations**) and the rule for the simulation envelope. Mostly you may use 199 (999) simulations and the 5th (25th) lowest and highest values as pointwise simulation envelopes.

2.3 Hints

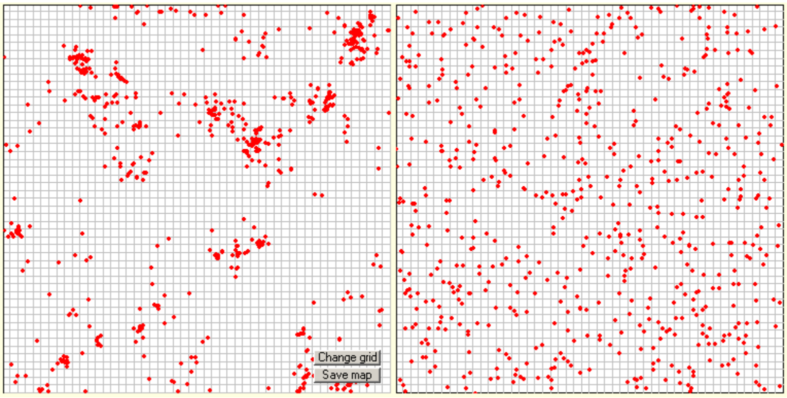
Almost all important elements on the screen contain a hint. For example, if you move the cursor over the element "**Load settings for Example**", a small message box shows the hint that briefly explains the meaning of the element.



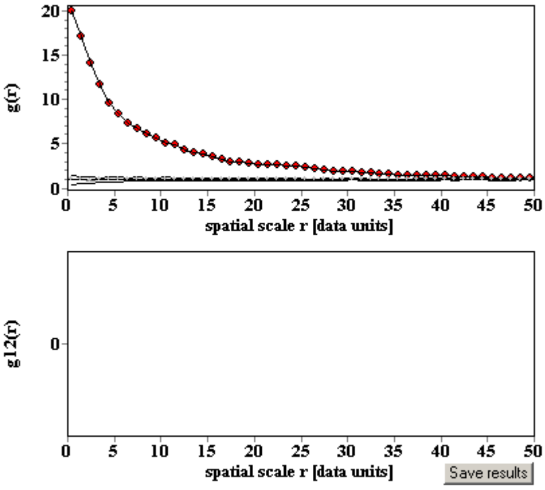
Here you can load the predefined settings for examples of point-pattern analysis

2.4 What happens on the screen?

After loading the settings file test.res as described above, *Programita* will automatically select all settings for the data and analysis mode and all settings for the null model. If you click "**Calculate Index**" *Programita* repeats the analysis. Two plots will appear additionally to the menus:
One plot shows the original point pattern being analyzed (left or top plot), and the other plot (on the right or bottom) shows the patterns of the Monte Carlo simulations of the null model used for constructing the simulation envelopes and the GoF test.

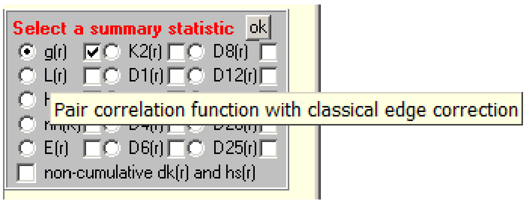


After the simulations of the null model are finished, the figure with the simulated patterns of the null model disappears, and instead a figure with the result of the analysis appears. The top (or left) figure shows generally the results of the univariate analysis and the bottom (right) figure shows the results of the bivariate analysis. (Exceptions are multivariate analysis using a dissimilarity matrix under the random labeling and trivariate random labeling mode). The data file in the example contained only one pattern (i.e., an univariate analysis), therefore no figure appears for the bivariate results



The observed summary functions are indicated by red dots, the pointwise simulation envelopes by black lines and the expectation of the null model by a grey dotted line.

In the standard mode you can then view the results of various summary functions, the hint explains the symbols.

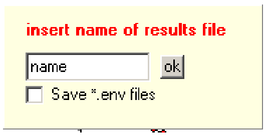

Select a summary statistic [ok]

● g(r) ☑ ○ K2(r) □ ○ D8(r) □
○ L(r) □ ○ D1(r) □ ○ D12(r) □
○ H
○ nn(k) □ ○ D4(r) □ ○ D20(r) □
○ E(r) □ ○ D6(r) □ ○ D25(r) □
□ non-cumulative dk(r) and hs(r)

Pair correlation function with classical edge correction

2.5 Save the results of the analysis

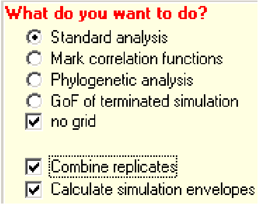
To save the results of an analysis, press the button "**Save results**" (it appears below the graph with the results of the bivariate analysis) and insert a name for the result file. The results file will be saved as ASCII-file name.res in the same directory where the *.exe file of *Programita* is located.


insert name of results file

name [ok]

□ Save *.env files

If the setting "**Combine replicates**" was enabled, *Programita* saves additionally the file **name.rep** that contains the information to recreate the results graph. Additionally, the *.rep files contain all information to combine the results of different replicate analysis. However, in the standard mode you need to save the results for each summary function separately.


What do you want to do?

● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

☑ Combine replicates
☑ Calculate simulation envelopes

The *.res results file for the standard analysis (see below) contains the settings of this analysis and the results of the univariate and the bivariate point pattern analysis. The results file name.res can be used to load the setting and to repeat the analysis. More importantly you can copy-paste the second part of the *.res file into a graphics program to produce the figures. The *.res files for the other modes of analysis look similar.

```
Pointpattern analysis of file c:\Programita\Book_Fig4_15a.dat
Method Wiegand-Moloney (ring) with  199 replicates for simulation envelopes, ring width = 3   5 th lowest and highest values of 199 simulations
Test Model= 12random
the null model assumed homogeneous pattern(s)
Analysis modus= points        gridless    WM    NN Hanisch  Hs factor   1.0
several points per cell allowed
All cells within the rectangle were considered for calculating the indices
number points of pattern 1 =  626
number points of pattern 2 =    0
the rectangular area contains 500*500 = 250000   cells  (= dim1*dim2)
 x-grid-size=  500 y-grid-size=  500 cell-size =  1.0000 units. rmax=     50,  max distance for NN functions:    354

Distancerr    g11(r)     E11-         E11+        Expect           g12(r)     E12-       E12+       Expect
   0.50 +r  19.9600000  0.4071000  1.7326000  1.0831684       0.0000000  0.0000000  0.0000000  0.0000000
   1.50 +r  17.0493000  0.5437000  1.6350000  1.0602789       0.0000000  0.0000000  0.0000000  0.0000000
   2.50 +r  14.0978000  0.6263000  1.4757000  1.0582842       0.0000000  0.0000000  0.0000000  0.0000000
   3.50 +r  11.6271000  0.8178000  1.3682000  1.0507368       0.0000000  0.0000000  0.0000000  0.0000000
   4.50 +r   9.5810000  0.7798000  1.3273000  1.0564684       0.0000000  0.0000000  0.0000000  0.0000000
   5.50 +r   8.3210000  0.8089000  1.2150000  0.9861684       0.0000000  0.0000000  0.0000000  0.0000000
   6.50 +r   7.2732000  0.8022000  1.1272000  0.9614842       0.0000000  0.0000000  0.0000000  0.0000000
   7.50 +r   6.6848000  0.8795000  1.1553000  0.9754263       0.0000000  0.0000000  0.0000000  0.0000000
   8.50 +r   6.0997000  0.8702000  1.1641000  1.0171474       0.0000000  0.0000000  0.0000000  0.0000000
   9.50 +r   5.6139000  0.8822000  1.1749000  1.0165895       0.0000000  0.0000000  0.0000000  0.0000000
```

The *.res results file (test.res). The **first 11 lines** contain the information on the settings of the analysis; the following part contains a table with the results of the analysis for the particular summary function selected. The important information for this example is heighted in grey. Note that the cell-size indicates in the standard analysis mode only the bin of the distance axis (given in data units), but in the grid-based standard analysis mode it indicates the size of the cells. In the following lines:

- The **first column** gives the spatial distance *r* of the point-pattern analysis in units of bins,
- the second and third column provide a summary of the Monte Carlo significance test of the null model ("-": data at scale *r* below the pointwise simulation envelopes, "r": inside the simulation envelopes, and "+": above the simulation envelopes; second column for univariate analysis, third column for bivariate analysis),
- **columns 4, 5, 6, 7**: results of univariate analysis (column 4: summary functions of the data, column 5: lower pointwise simulation envelope, column 6: upper pointwise simulation envelope, column 7: expectation under the null model),
- **columns 8, 9, 10, 11:** results of bivariate analysis (column 8: summary functions of the data, column 9: lower simulation envelope, column 6: upper simulation envelope, column 7: expectation under the null model).

2.6 Simulation envelopes, Goodness-of-Fit (GoF) and global tests

2.6.1 Simulation envelopes

An important step of all methods of point pattern analysis implemented in *Programita* is the comparison of the observed pattern with patterns generated by stochastic point process or null models. The objective of this comparison is generally to examine ecological hypotheses.

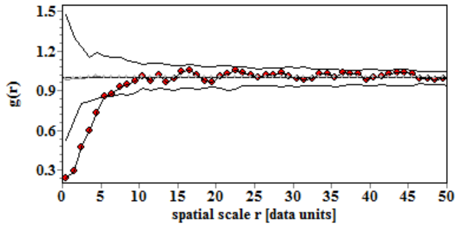
Spatial point process models are mathematical models that provide a stochastic mechanism to generate point patterns. The spatial structure in a point process model is usually governed by a set of parameters (e.g., the degree of clustering), which must be fitted to the observed pattern. Point process models are typically used to

- describe the data as close as possible and to summarize the statistical properties of the observed point patterns with few parameters
- represent the expected spatial pattern according to specific ecological hypotheses that are being tested.

Null models are a subclass of point process models and formalize a particular null hypothesis in ecology. Basically, null models create the spatial patterns that are expected in the absence of a particular ecological mechanism by means of the randomization of ecological data where certain elements of the data are held constant, and others are allowed to vary stochastically. The null model is therefore used to determine whether there is spatial structure in the data that does not exist in the null model. Although this approach is often not very informative for univariate patterns, it can be useful for detecting spatial structure in more complex data types which are often extremely relevant for ecologists.

Monte Carlo methods are used to generate multiple realizations of the null model or point process (e.g., 199 or 999 replicate patterns) to be compared to the observed data. To this end, the summary functions $S_0(r)$ of the observed pattern and of each of the *i* null model patterns $S_\mathrm{i}(r)$ are estimated. **The task is to find out if the summary functions of the observed pattern fall outside the typical range of the patterns produced by the model.**

Pointwise simulation envelopes are mostly used in ecology. They are for example the 5th lowest and highest values of the pair correlation functions of 199 simulated patterns at distance *r*, which yield a significance level of α = 0.05 for a fixed distance *r*. If the observed summary function lies at some distance *r* outside the pointwise simulation envelopes it is often taken as evidence of a departure from the null hypothesis.



Pointwise simulation envelopes of the example GoF.res based on the 5th lowest and highest values of 199 simulations of the CSR null model

The figure indicates that the pattern of the data set HC_2.dat shows regularity because the observed pair correlation function *g*(*r*) is at distances 0.5 – 5.5 m clearly below the pointwise simulation envelopes.

The graphical representation of pointwise simulation envelopes is especially attractive for ecological applications because it encircles the fluctuations of the summary function under the null model and points to distances where departures may occur.

However, departures of the observed summary function from the pointwise simulation envelopes cannot be used to reject the null model with significance level α because this analysis typically conducts multiple tests, one at each spatial distance bin *r*, and is prone to type I error inflation (Loosmore and Ford 2006, Wiegand et al. 2016). More refined methods are required for this purpose. Methods implemented in *Programita* include

- **the Goodness-of-Fit test** promoted by Loosmore and Ford (2006) that appeared also in the classical book of Diggle (1983, 2003) and was used before by ecologists (Velázquez et al. 2016b). This test collapses the scale-dependent information of the summary function *S*(*r*) into one test function that represents the accumulated squared deviation of the observed summary function from the expected summary function under the null model. This test does not lead to simulation envelopes but estimates a *P*-value that corresponds to the prescribed significance level *α*.
- Maximum Absolute Deviation (MAD) tests introduced by Diggle (1979) and Ripley (1979) reduce the multiple tests at different distance bins *r* into a single test statistic being the maximum absolute value of the difference between the summary function expected under the null model and the summary function of the data, taken over all distance bins. This test leads to simulation envelopes of constant width, centred on the expected summary function. **However, the envelopes arising from the simple MAD test have a major problem**: they are not sufficiently flexible to represent the behaviour of the summary functions of the null model for different distances *r* if their distribution of is not the same for all *r*. They are therefore not implemented in *Programita*.
- **MAD tests of studentized summary functions.** To overcome the problem of the simple MAD test, Wiegand et al. (2016) and Myllymäki et al. (2015, 2017) proposed to transform first the summary functions (to obtain at each distance bin *r* an identical distribution of the values of the summary functions of the null model simulations) before applying the MAD test. They used for this purpose the so-called studentization of the summary functions that produces basically standardized effect sizes (also called z-scores) of the summary functions by subtracting the mean and dividing by the standard deviation. The resulting simulation envelopes are constant and have therefore the desired property that the null hypothesis can be rejected with significance level α if the empirical summary function wanders outside the envelopes.
- **maximal global envelope tests** first studentize the summary functions, estimate from this the simulation envelopes of the MAD test (that show a the prescribed significance level *α*) and then use the inverse student transformation of the MAD envelopes to obtain the **global simulation envelopes**. The term “global” indicates that the significance level *α* of the test is valid for a whole given distance interval and not only for one distance *r* as for the pointwise simulation envelopes. The global simulation envelopes have therefore the desired property that the null hypothesis can be rejected with significance level α if the empirical summary function wanders outside the global envelopes.

2.6.2 Goodness-of-Fit test

The pointwise simulation envelopes provide a good idea on the range of the summary functions expected under the null model, but they cannot be used to assess the general fit of the model because of problems associated with Type I error inflation. This can be avoided by one of the methods listed above, for example, by using additionally a Goodness-of-Fit test (GoF) (see Loosmore and Ford 2006 and section 2.5.1.2 of Wiegand and Moloney 2014) or a global envelope test (see Wiegand et al. 2016).

A GoF is used to test if a given null model (or point process model) fits a given summary function of the observed data over a given distance interval ($r_{\min}$, $r_{\max}$). (However, the test is conservative if the point process model involved fitting of parameters). The GoF test collapses the scale-dependent information of a functional summary function [e.g., *g*(*r*)] into a single index u_{i}. The index u_{i} represents the accumulated squared deviation of the observed summary function from the expected summary function under the null model, summed up over an appropriate distance interval ($r_{\min}$, $r_{\max}$) (Loosmore and Ford 2006):

$$u_i = \sum_{r=r_{\min}}^{r_{\max}} (\hat{S}_i(r) - S(r))^2$$

where the $\hat{S}_i(r)$ is the empirical summary function of the observed pattern (i = 0) and that of the simulated patterns (i = 1,...*m*), and *S*(*r*) the expected summary function under the null model. If the expected summary function *S*(*r*) is not known analytically, it can be replaced by

$$\overline{S}_i(r) = \frac{1}{m} \sum_{j=0, j\neq i}^{m} \hat{S}_j(r)$$

which is the average over all summary functions, $\hat{S}_i(r)$ except the one with index i. Note that $\overline{S}_0(r)$ yields the average over the summary function of all *m* simulated patterns and provides therefore an unbiased estimate of *S*(*r*) under the null model. For the GoF test the u_{i} are calculated for the observed data (i = 0) and for the simulated data (i = 1...*m*) and the rank of u_0 among all u_{i} is determined. The observed *P*-value of this test is

$$\hat{p} = 1 - \frac{rank[u_0] - 1}{m+1}$$

For example, if the u_0 computed for the observed pattern was larger than the u_{i} computed for each of the *m* = 199 simulations of the null model we have rank[u_0] = 200 and $\hat{p} = 1 - (199/200) = 0.005$.

Note that this GoF test is somewhat sensitive to the distance interval selected. For example, if the departure from the null model occurs only at small scales of say 5m, but the test is conducted over an interval of 0–100m a true departure may be overpowered and not detected. Therefore use only an interval where departures from the null model are (a priori) expected. The P-value alone does not always convey the nature of discrepancy between the data and the null model. It should always be used in conjunction with visual inspection of the pointwise simulation envelopes.

2.6.3 Global simulation envelopes

Global envelopes $S^+(r)$ and $S^-(r)$ that are variable in *r* were proposed by Wiegand et al. (2016) and Myllymäki et al. (2017). They have the desired and intuitive property that the null model can be rejected over a given distance interval with significance level *α* if the observed summary function *S*(*r*) wanders at one or more distances *r* outside the global simulation envelopes. Pointwise envelopes do not have this property because of the problem of multiple testing (Loosmore et al. 2006). Myllymäki et al. (2017) presented a version of the test that is based on simulations and can be applied generally whereas Wiegand et al. (2016) presented an analytical version of the test that applies for non-cumulative summary functions such as the pair correlation function.

The global envelopes $S^+(r)$ and $S^-(r)$ are constructed in three steps. First, the summary functions $S_i(r)$ are estimated from the observed data (*i* = 0) and from the *m* realizations of the null model (*i* = 1,.. *m*), and the mean $\overline{S}(r)$ and the standard deviation $\hat{\sigma}_S(r)$ of the $S_i(r)$ are estimated for *i* = 1,.. *m*. Then, the original summary functions $S_i(r)$ are student transformed:

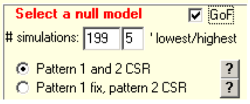
$$S_i^{ses}(r) = \frac{S_i(r) - \overline{S}(r)}{\hat{\sigma}_S(r)}, \; i = 0, \dots, m.$$

Notably, the pointwise simulation envelopes $G_\mathrm{p}^-(r)$ and $G_\mathrm{p}^+(r)$ of the student transformed summary functions (e.g., for $\alpha = 0.05$ the 5th lowest and highest values of $S_i^{\mathrm{ses}}(r)$ taken from *i* = 1, .., 199) approximate for all distances *r* the critical value $G_\mathrm{p}^-(r) = -z_\alpha$ and $G_\mathrm{p}^+(r) = z_\alpha$ with $z_\alpha = 1.96$ for $\alpha = 0.05$. Thus, we have constant pointwise simulation envelopes. This works if the distribution of the $S_i(r)$ for *i* = 1, …*m* approximates for fixed values of *r* a normal distribution. **This assumption can be tested** by comparing the $G_\mathrm{p}^-(r)$ and $G_\mathrm{p}^+(r)$ with the critical values z_α and $-z_\alpha$. If the distribution is not symmetric for some values of *r* one can either use **upper and lower quantiles** proposed by Myllymäki et al. (2017) or exclude these distances from the distance interval where the global envelope test is applied.

Second, the standard "maximal absolute difference" (MAD) test is applied for the studentised summary functions $S_i^{\mathrm{ses}}(r)$. This test makes sense now because the variance of the $S_i^{\mathrm{ses}}(r)$ under the null model is the same for all distances *r*. The functional summary function $S_i^{\mathrm{ses}}(r)$ of the *i*th simulation of the null model is reduced to its minimum and maximum value S_i^{min} and S_i^{max}, respectively, taken over the distance interval $r = r_{\mathrm{min}}, .., r_{\mathrm{max}}$ of interest. The *k*th largest value of the S_i^{max} is the upper global envelope G^+, and the *k*th smallest value of the S_i^{min} is the lower global envelope G^-. Note that this test conducts only one test for the entire interval. For this reason, the problem of multiple inference (Loosmore et al. 2006) does not occur and we can reject the null model with significance level *α* if $S_0^{\mathrm{ses}}(r) > G^+$ or $S_0^{\mathrm{ses}}(r) < G^-$ for one or more distances *r* ($r \geq r_{\mathrm{min}}$ and $r \leq r_{\mathrm{max}}$).

Third, to obtain the desired global simulation envelopes $S^+(r)$ and $S^-(r)$ that are variable in *r* we apply the inverse student transformation to G^+ and G^- (see eq. 17 in Myllymäki et al. 2017):

$$S^+(r) = \overline{S}(r) + \hat{\sigma}_S(r) G^+$$
$$S^-(r) = \overline{S}(r) - \hat{\sigma}_S(r) G^-$$

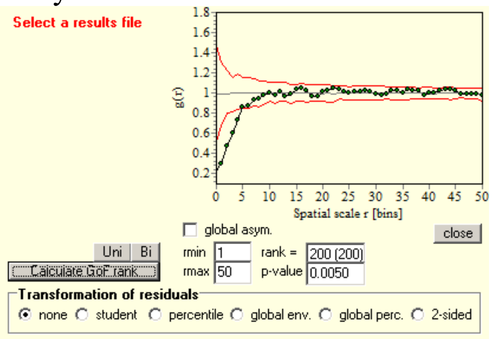


# simulations: 199 5 ' lowest/highest

● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR

Programita allows you to access the GoF and global envelope tests in two ways. First, after termination of a simulation of the null model a small checkbox "**GoF**" appears top right on the window "Select a null model". After enabling the check box a window appears. Click "**Uni**" or "**Bi**", depending if the analysis of interest is uni- or bivariate, respectively. Then a small graph with the observed summary function and the lowest and highest values of the null model appear.

Provide now the distance interval (rmin, rmax) to be tested and click "**Calculate GoF rank**". The rank and the associated P-value of the GoF test are then provided. The figure is updated and shows now the pointwise simulation envelopes with the nth lowest and highest values of the summary function of the null model simulations at distance *r*.



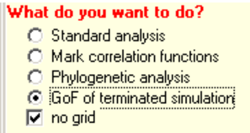
□ global asym.

rmin 1 rank = 200 (200)
rmax 50 p-value 0.0050

Transformation of residuals
● none ○ student ○ percentile ○ global env. ○ global perc. ○ 2-sided

Programita saves for the selected test statistic temporary files Uni_confidence.env (or Bi_confidence.env). This file contains the observed uni- or bivariate summary function of the data (first line) and of all simulations of the null model (following lines).

The second way to access the GoF test is to click in the menu "What do you want to do?" the option "**GoF of terminated simulation**". With this option *Programita* reads a *.env file that you saved after running a simulation with "**Save results**".


What do you want to do?
○ Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
● GoF of terminated simulation
☑ no grid

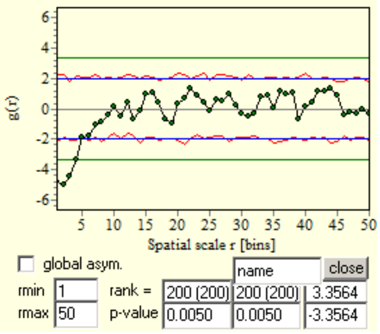
Additional options: student transformation and pointwise envelopes

The power of the GoF test can be enhanced by transforming the summary function (Myllymäkia et al. 2015). If the distribution of a summary function *S*(*r*) estimated for the simulations of the null model at distance *r* approximates a normal distribution with mean $\bar{S}(r)$ and standard deviation $\hat{\sigma}_S(r)$ we can apply the student transformation

$$S_i^{ses}(r) = [S_i(r) - \bar{S}(r)] / \hat{\sigma}_S(r)$$

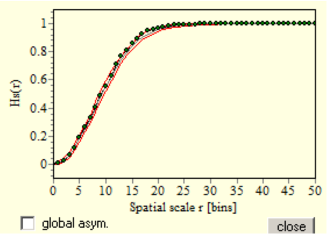
which transforms the raw values $S_i(r)$ of the summary function into a "standardized effect size" $S_i^{ses}(r)$ (sometimes also called z-score). If we consider only one distance *r*, we have for $S_i^{ses}(r) > 1.96$ a significant positive departure with significance level $\alpha = 0.05$, for $S_i^{ses}(r) <$ -1.96 a significant negative departure with $\alpha = 0.05$, and the larger the absolute value $S_i^{ses}(r)$ the stronger the departure from the null model.

The option ● student uses the $S_i^{ses}(r)$ to conduct the GoF test and a MAD test (see below). The figure shows the student-transformed observed $S_0^{ses}(r)$ (black dots), the pointwise simulation envelopes estimated from $S_i^{ses}(r)$ (red lines), the theoretical value for a 5% significance level (i.e., 1.96; blue line). The pointwise simulation envelopes have now at all distances *r* approximately the same width! The left values for the rank and the p-value are for the GoF test based on the $S_i^{ses}(r)$. The file SES_name.res (where name is the name of the *.dat data file or a name you provide) shows the data for plotting the figure on the right together with $\bar{S}(r)$ and $\hat{\sigma}_S(r)$.

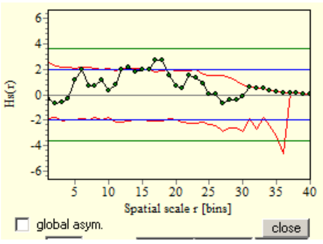


□ global asym. name

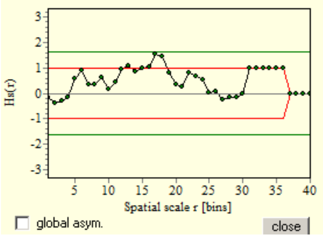
rmin 1 rank = 200 (200) 200 (200) 3.3564
rmax 50 p-value 0.0050 0.0050 -3.3564



student transformation



percentile transformation



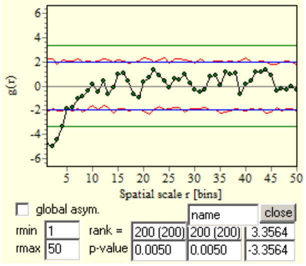
Please check that the red lines (i.e., the simulation envelopes of the transformed $S^{ses}(r)$ approximate the theoretical 5% value of 1.96 (as in the example). If the $S(r)$ of the null model do not approximate the normal distribution (e.g., if the residuals are not symmetric) you will observe a bias in the simulation envelopes as in the example of the spherical contact distribution for distances $r > 22$. This is because for $r > 22$ many of the $H_S(r)$ values show the maximal value of one which yields a non-symmetric distribution. In this case conduct the GoF test only for the interval where simulation envelops (red lines) approximate the theoretical value (blue line). Alternatively use the percentile transformation (see below).

Additional options: student transformation and MAD test

The effect of the student (or percentile) transformation is to homogenize the residuals which make the pointwise simulation envelopes basically a constant. If the width of the simulation envelopes is the same for all distances *r*, it makes sense to construct a "global envelop" test (Ripley 1981) based on the maximal absolute deviation of the observed summary function from the expectation. This test then reduces the functional summary functions into one or two numbers: its minimal and maximal value taken over the interval rmin to rmax of interest.

If you check ☑ global asym. this test reduces the functional summary function $S_\text{i}^{ses}(r)$ of the *i*th simulation of the null model into two numbers, its minimal and maximal value S_i^{min} and S_i^{max}, respectively, taken over the interval rmin to rmax. The upper global envelop G^+ is then the *k*th largest value of the S_i^{max} and the lower global envelop G^- is the *k*th smallest value of the S_i^{min}.

If you do not check □ global asym. this test reduces the functional summary function $S_i^{ses}(r)$ of the ith simulation of the null model into one numbers, the maximal value S_i^{max} of $\|S_i^{ses}(r)\|$, taken over the interval rmin to rmax. The upper global envelop G^+ is the 2**k*th largest value of the S_i^{max}, and the lower global envelop $G^+ = - S_i^{max}$. In the example we find $G^+ = 3.356$. The green lines now show the upper and lower global envelopes. The right values for the rank and the p-value are for the global test based on $S_i^{gl} = \max(\|S_i^{max}\|, (\|S_i^{min}\|)$ where the rank of S_0^{gl} among all S_i^{gl} is determined.



rmin 1 rank = 200 (200) 200 (200) 3.3564

rmax 50 p-value 0.0050 0.0050 -3.3564

The file "transformENV_st.env" shows the values of the transformed summary functions $S^{ses}(r)$, the $\overline{S}(r)$ and $\hat{\sigma}_S(r)$, the S_i^{min} and S_i^{max} and lower and upper global envelopes G^+ and G^-.

The file SES_name.res (where name is the name of the *.dat data file or a name you provide) shows the data for plotting the figure on the right together with $\overline{S}(r)$ and $\hat{\sigma}_S(r)$ and the file envelope.env gives the 1 + m transformed summary functions.

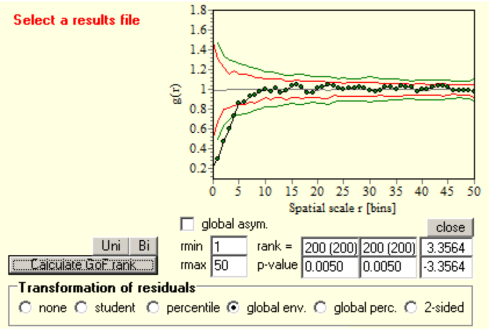
Additional options: global simulation envelopes that are variable in *r*

The global test with effect size scaling allows for an inverse transformation of the global envelopes S^{+}_{ses} and S^{-}_{ses} (if the distributions of $S_i^{\text{ses}}(r)$ follow a standard normal distribution) to yield the corresponding envelops $S^{+}(r)$ and $S^{-}(r)$ with the exact α value in the non-scaled representation:

$$S^{+}(r) = G^{+}\hat{\sigma}_S(r) + \bar{S}(r)$$

$$S^{-}(r) = G^{-}\hat{\sigma}_S(r) + \bar{S}(r)$$

This allows for estimation of global envelopes in the non-transformed representation of the summary function *S*(*r*) if you click the option ● global env. . The green lines show the global envelopes, and the red lines the pointwise envelopes (i.e., the nth lowest and highest value of $S_i(r)$). The left rank and p-value is GoF test for the non-transformed envelop test (option "none") and the right rank and p-value is for the global envelope test of the student transformed $S_i(r)$. The file GL_name.res provides the data to plot the figure together with $\hat{\sigma}_S(r)$ and the file envelope.env gives the 1 + *m* transformed summary functions.



Select a results file

□ global asym. close

Uni Bi

Calculate GoF rank

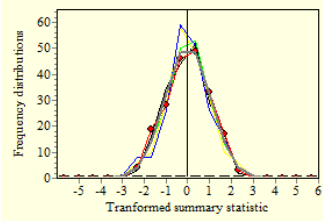
rmin 1 rank = 200 (200) 200 (200) 3.3564

rmax 50 p-value 0.0050 0.0050 -3.3564

Transformation of residuals

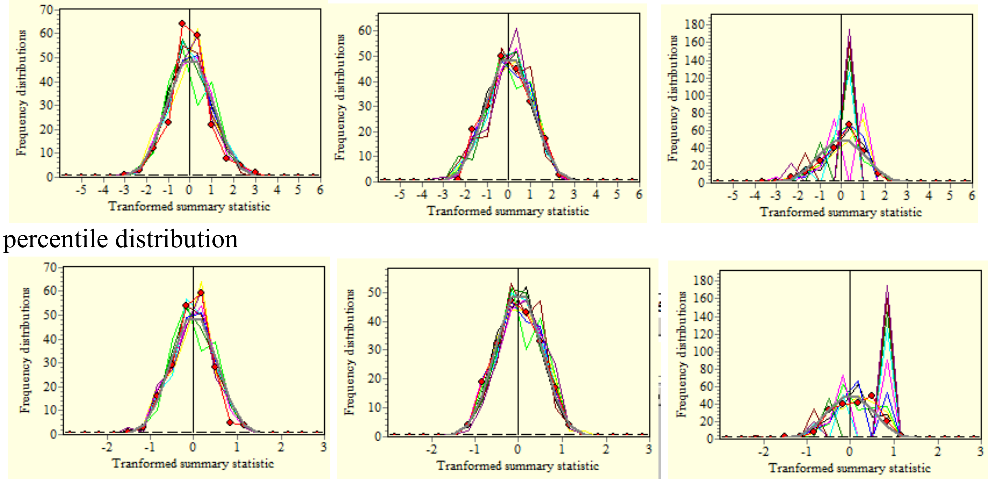
○ none ○ student ○ percentile ● global env. ○ global perc. ○ 2-sided

Programita allows you to check the assumption that the distributions of $S_i^{\text{ses}}(r)$ follow a standard normal distribution. Check ☑ check distributions and press again Calculate GoF rank and Programita shows you the distributions of the $S_i^{ses}(r)$ (taken for *i* = 1,...m) for fixed values of *r*. It shows the distribution for 10 values of *r* that a regularly spaced over rmin – rmax. In the example *r* = 1– 5. The grey bold line shows the standard normal distribution For the example of the *g*(*r*) the $S_i^{\text{ses}}(r)$ follow the assumption well. However, as shown above, this assumption does not hold for the spherical contact distributions $H_s(r)$ shortly before saturation (where all values are close to 1). Below see the distributions for different intervals of *r*. For *r* > 25 the distribution becomes asymmetric.



r = 1–15   *r* = 15–25   *r* = 25–35

student transformation

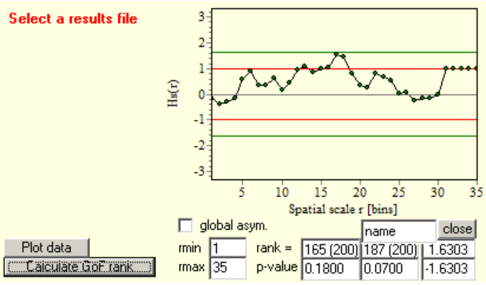


percentile distribution

Additional options: percentile scaling

The student transformation works only if the $S_i(r)$ are approximately normally distributed (symmetric). However, some summary functions which are bounded to minimal and maximal values (such as $H_s(r)$ may show non-symmetric residuals and in this case the global envelopes of the student transformed summary functions $S^{ses}(r)$ may show a bias. One possibility to avoid this is to scale with the upper and lower simulation envelopes $S^+(r)$ and $S^-(r)$ and the expectation of the null model $\bar{S}(r)$:

$$S_i^{perc}(r) = \begin{cases} \dfrac{S_i(r)-\bar{S}(r)}{S^+(r)-\bar{S}(r)} & S_i(r) > \bar{S}(r) \\[2ex] \dfrac{S_i(r)-\bar{S}(r)}{S^-(r)-\bar{S}(r)} & S_i(r) < \bar{S}(r) \end{cases}$$


Select a results file

☐ global asym.

name close

Plot data

Calculate GoF rank

rmin 1

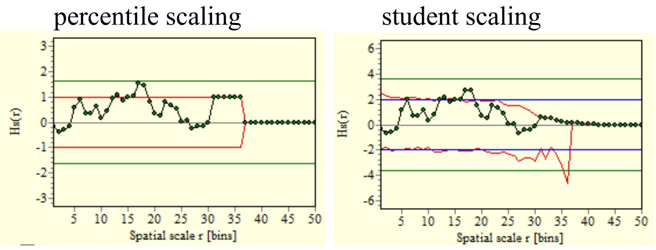
rmax 35

rank = 165 (200) 187 (200) 1.6303

p-value 0.1800 0.0700 -1.6303

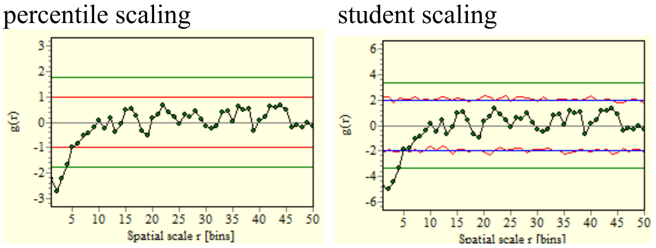
The file "PERC_name.res" shows the data to plot the figure on the right, and the file envelope.env gives the 1 + m transformed summary functions.

The figures on the right show the difference between percentile and student scaling for the asymmetric distributions of the nearest neighbor distance distribution $H_s(r)$.


percentile scaling

student scaling

The figures on the right show that there is basically no difference between percentile and student scaling for the symmetric distributions of the pair correlation function $g(r)$.


percentile scaling

student scaling

Standard T-test:

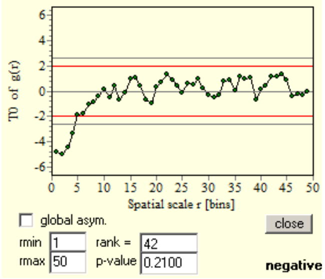
The GoF test presented above is based on a one-sided test statistics. A similar two-sided test statistics can be constructed based on the standard T-test (Diggle et al. 2007; equation 2.10 in Wiegand and Moloney 2014). You can access this test with the option ○ 2-sided.

This test uses, similarly to the student transformation, the variables $S_i(r)$, the summary function for the observed data (for i = 0) and the summary functions of the m simulations of the null model (for i = 1 to m), and the variables $\bar{S}(r)$ and $\hat{\sigma}_S(r)$ which are the mean and standard deviation of the summary function of the m simulations of the null model at distance r. The test is conducted over an appropriate distance interval ($r_{\min}$, $r_{\max}$) where departures from the null model are expected (before conducting the test). For the data (i = 0) and the m simulations of the null model (for i = 1 to m) the test statistic.

$$T_i = \sum_{r=r_{\min}}^{r_{\max}} \left(\frac{S_i(r) - \overline{S}(r)}{\hat{\sigma}_S(r)}\right) = \sum_{r=r_{\min}}^{r_{\max}} Z_i(r)$$

is estimated. It is the sum of the standardized values $Z_i(r)$ over the selected distance interval. The transformation $Z_i(r)$ of the original summary functions $S_{\mathrm{i}}(r)$ is a so-called “studentised scaling”. At each distance bin r the values 1.96 and 1.96 are simulation envelopes with approximately 5% error rate if the values of $S_{\mathrm{i}}(r)$ (i= 1,.., m) follow a normal distribution with mean $\overline{S}(r)$ and standard deviation $\hat{\sigma}_S(r)$.

The p-value of this GoF test is obtained by estimating T_0 from the data and comparing it to each of the m estimated T_{i} statistics from the null model. The significance level of the test is given by $\boldsymbol{P = (k + 1)/(m + 1)}$, where k is the number of simulated T_{i} greater (or smaller) than T_0 if the departure was positive (or negative). Access the GoF test as described above. A small graph with the observed summary function and the lowest and highest values of the null model appear. Provide now the distance interval (rmin, rmax) to be tested and click “**Two-sided test**”.



□ global asym.

rmin 1 rank = 42

rmax 50 p-value 0.2100

negative

Programita now estimates the $Z_0(r)$ values of the observed data

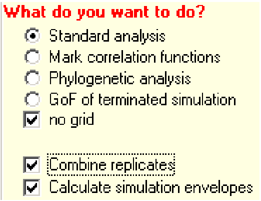
$$Z_0(r) = \frac{S_0(r) - \overline{S}(r)}{\hat{\sigma}_S(r)}$$

for the selected distance interval and plots the values of $Z_0(r)$ together with the critical bands for a P-value of 0.05 (red) and 0.01 (grey). The rank, the associated P-value of the T_{i} test over distance interval ($r_{\min}$, $r_{\max}$) and the direction of the departure (i.e., positive or negative) are then provided. The red lines corresponds to 5% envelopes (1.96) and the grey line to 1% envelopes (2.575).

Note that the $Z_{\mathrm{i}}(r)$ test statistic can have negative and positive values and if the observed summary function shows at some distances positive and at others negative departures from the null model, they may cancel.

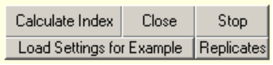
2.7 Show results of previous analyses

2.7.1 Show results of previous analyses: standard analysis



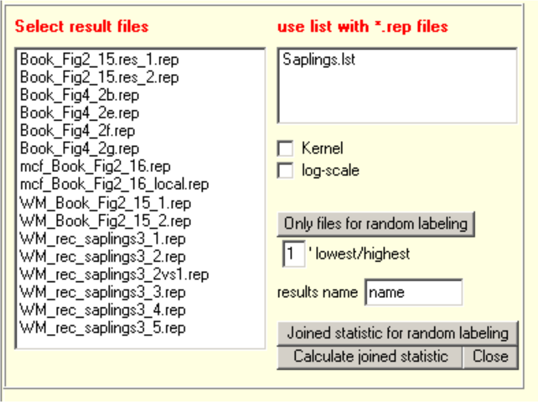
What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

☑ Combine replicates
☑ Calculate simulation envelopes



Programita offers a convenient possibility to show the results of previous analyses. In the standard analysis mode, this works only if the option "**Combine replicates**" was enabled when doing the original analysis. However, for mark correlation analyses and multivariate (phylogenetic) analyses this mode is automatically enabled.

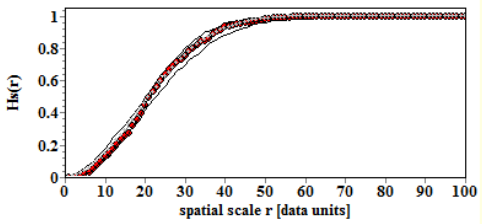
To show the results of a previous analysis, apply the button "**Replicates**" and a window with a list of results files appears:



Select result files
Book_Fig2_15.res_1.rep
Book_Fig2_15.res_2.rep
Book_Fig4_2b.rep
Book_Fig4_2e.rep
Book_Fig4_2f.rep
Book_Fig4_2g.rep
mcf_Book_Fig2_16.rep
mcf_Book_Fig2_16_local.rep
WM_Book_Fig2_15_1.rep
WM_Book_Fig2_15_2.rep
WM_rec_saplings3_1.rep
WM_rec_saplings3_2.rep
WM_rec_saplings3_2vs1.rep
WM_rec_saplings3_3.rep
WM_rec_saplings3_4.rep
WM_rec_saplings3_5.rep

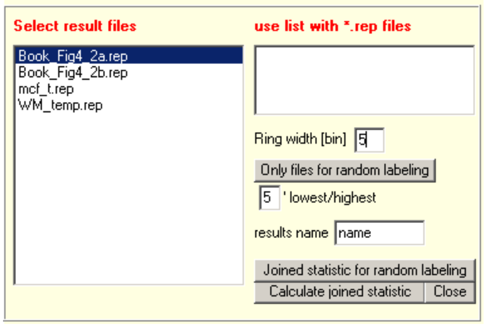
use list with *.rep files
Saplings.lst
□ Kernel
□ log-scale
Only files for random labeling
1 ' lowest/highest
results name: name
Joined statistic for random labeling
Calculate joined statistic
Close

Highlight for example Book_Fig4_2b.rep to repeat the analysis of Figure 4.2b in Wiegand and Moloney (2014), and click "**Calculate joined statistic**". The result of the analysis will then appear:



Change ring width for pair correlation function

If you use the pair correlation function as summary function, this feature of *Programita* allows you also to change a posteriori the ring width. **To take advantage of this feature, the original analysis must be done with ring width of 1**: 1 ring widht . For ring width = 1 the plot of the pair correlation function will be rugged.

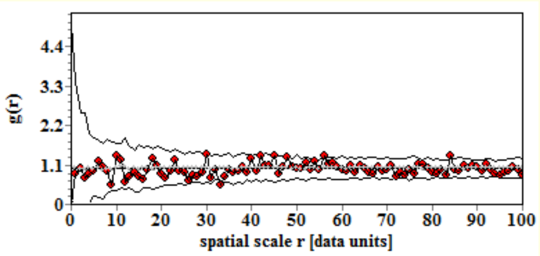


Select result files
Book_Fig4_2a.rep
Book_Fig4_2b.rep
mcf_t.rep
WM_temp.rep

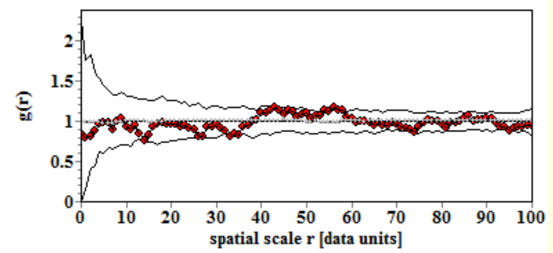
use list with *.rep files
Ring width [bin] 5
Only files for random labeling
5 ' lowest/highest
results name: name
Joined statistic for random labeling
Calculate joined statistic
Close

To select a posteriori a wider ring width select the file "Book_Fig4_2b.rep" to read the results of the analysis of Figure 4.2a in Wiegand and Moloney (2014), and then go" again to "Replicates". Now, when highlighting the again the file "Book_Fig4_2b.rep" you can select a new ring width, for example a ring width dr = 5: Ring width [bin] 5 . See below the example with dr = 1 (left) and dr = 5 (right):

dr = 1



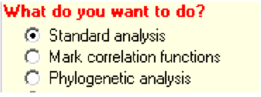
dr = 5

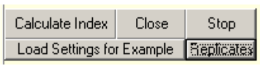


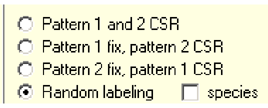
See also the description of combine replicates. Clearly, if you "combine" only the results of one "replicate", you view the analysis of one analysis.

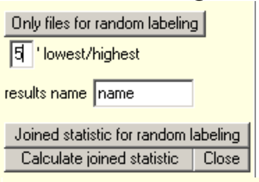
2.7.2 Show results of previous analyses: random labeling

To show the results of a previous analysis, apply the button “**Replicates**” and a window with a list of results files appears. If the null model in a standard analysis was random labeling, *Programita* saved two results files per analysis. To simplify selection of results files and to tell *Programita* that you will combine replicates that used random labeling click “**Only files for random labeling**”.

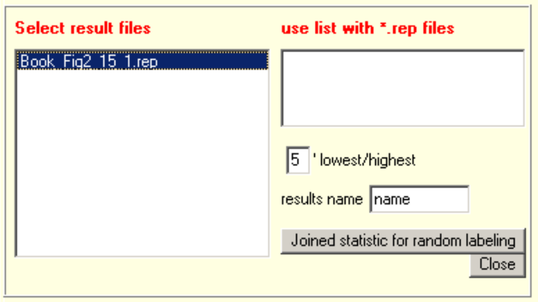

What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis



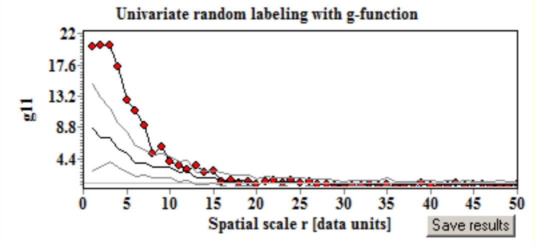

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
● Random labeling □ species


Only files for random labeling
Joined statistic for random labeling
Calculate joined statistic
Close
5 ' lowest/highest
results name name

Then you can select the *_1.rep file that was saved from your previous analysis:

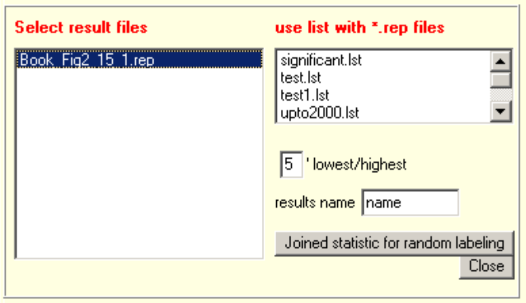

Select result files
Book_Fig2_15_1.rep
use list with *.rep files
5 ' lowest/highest
results name name

Highlight for example Book_Fig2_15_1.rep to repeat the analysis of Figure 2.15 in Wiegand and Moloney (2014), and click “**Calculate joined statistic**”. The result of the analysis shows up:



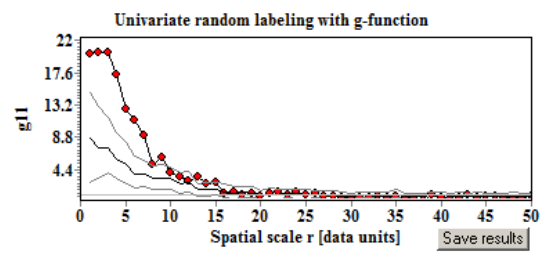
Change ring width for pair correlation function

If you use the pair correlation function as summary function, this feature of *Programita* allows you also to change a posteriori the ring width. **To take advantage of this feature, the original analysis must be done with ring width of 1**: 1 ring widht . For ring width = 1 the plot of the pair correlation function will be rugged.

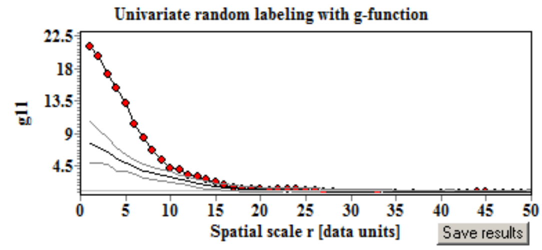

Select result files
Book_Fig2_15_1.rep
use list with *.rep files
significant.lst
test.lst
test1.lst
upto2000.lst
5 ' lowest/highest
results name name

To select a posteriori a wider ring width select the file “Book_Fig2_15_1.rep” to read the results of the analysis of Figure 2.15 in Wiegand and Moloney (2014), and then go” again to “Replicates”. Now, when highlighting the again the file “Book_Fig2_15_1.rep” you can select a new ring width, for example a ring width dr = 5: Ring width [bin] 5 . See below the example with dr = 1 (left) and dr = 5 (right):

dr = 1



dr = 5

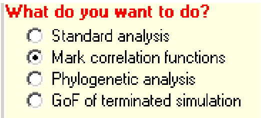


See also the description of combine replicates. Clearly, if you “combine” only the results of one “replicate”, you view the analysis of one analysis.

2.7.3 Show results of previous analyses: mark correlation functions

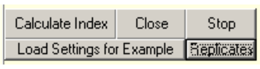
You can also change *a posteriori* the ring width, use the cumulative or the non-normalized mark correlation function.

To take advantage of these features, the original analysis must be done with ring width of 1: ☐1 ring widht.

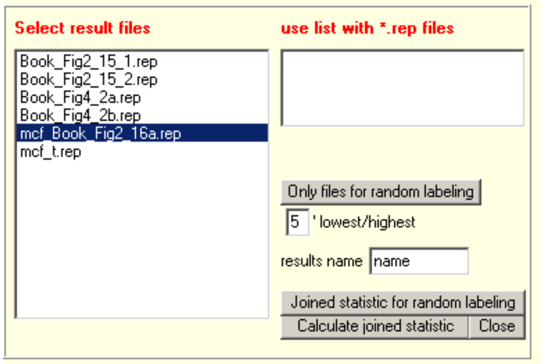


What do you want to do?

- ○ Standard analysis
- ● Mark correlation functions
- ○ Phylogenetic analysis
- ○ GoF of terminated simulation



To show the results of a previous mark correlation analysis, apply the button “**Replicates**” and a window with a list of results files appears:



Select result files

Book_Fig2_15_1.rep
Book_Fig2_15_2.rep
Book_Fig4_2a.rep
Book_Fig4_2b.rep
mcf_Book_Fig2_16a.rep
mcf_t.rep

use list with *.rep files

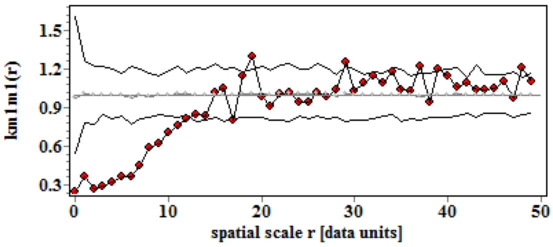
Only files for random labeling

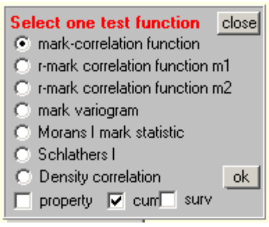
5 ' lowest/highest

results name: name

Joined statistic for random labeling
Calculate joined statistic
Close

Note that the *.rep files of the mark correlation analysis show also the prefix “mcf_”. Highlight for example mcf_Book_Fig2_16a.rep to repeat the analysis of Figure 2.16 in Wiegand and Moloney (2014), and click “**Calculate joined statistic**”. The result of the analysis shows up:



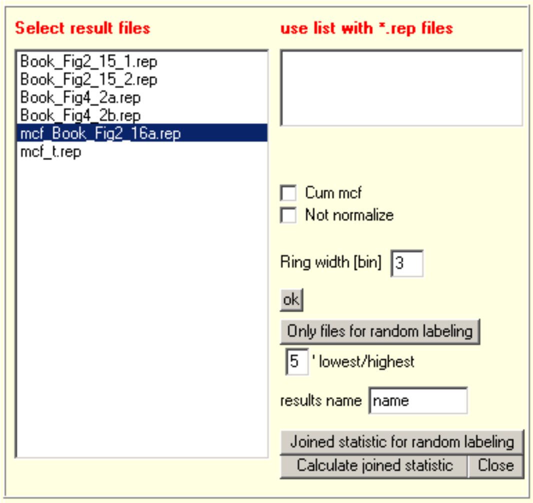


Select one test function

- ● mark-correlation function
- ○ r-mark correlation function m1
- ○ r-mark correlation function m2
- ○ mark variogram
- ○ Morans I mark statistic
- ○ Schlathers I
- ○ Density correlation

☐ property ☑ cum ☐ surv

In the box “Select one test function” you can now select the different mark correlation functions implemented in *Programita*.



Select result files

Book_Fig2_15_1.rep
Book_Fig2_15_2.rep
Book_Fig4_2a.rep
Book_Fig4_2b.rep
mcf_Book_Fig2_16a.rep
mcf_t.rep

use list with *.rep files

☐ Cum mcf
☐ Not normalize

Ring width [bin] 3

ok
Only files for random labeling

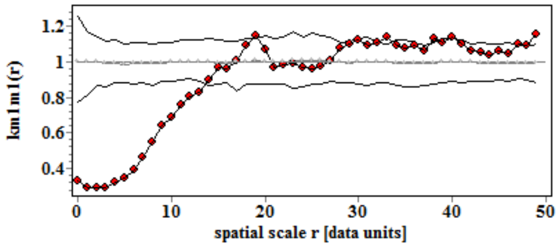
5 ' lowest/highest

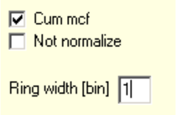
results name: name

Joined statistic for random labeling
Calculate joined statistic
Close

Change ring width for non-cumulative functions

To select a posteriori a wider ring width enter now a new width dr = 3, and click “the small “ok” button:



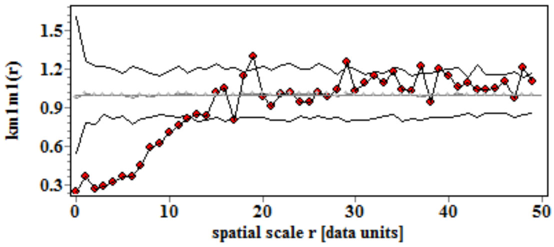


☑ Cum mcf
☐ Not normalize

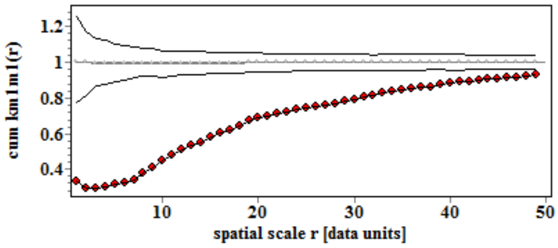
Ring width [bin] 1

To obtain the corresponding cumulative mark correlation function, select “Cum mcf”, a ring width of 1 and click “the small “ok” button:

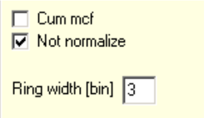
con-cumulative, dr=1



cumulative



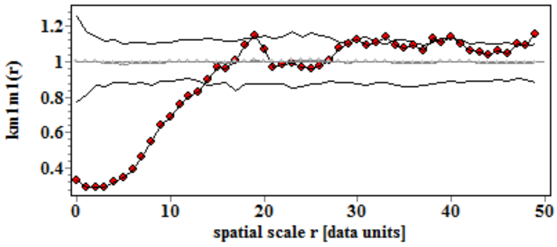
Show non-normalized mark correlation functions



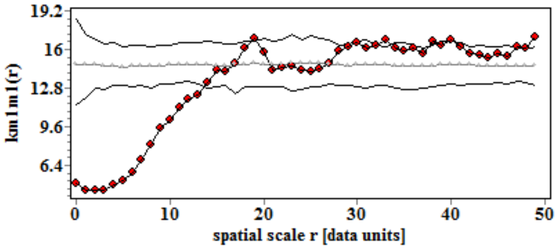
☐ Cum mcf
☑ Not normalize
Ring width [bin] 3

To obtain the corresponding non-normalized mark correlation function select “Not normalized”, a ring width of your choice (e.g., dr = 3), and click “the small “ok” button:

dr = 3, non-cumulative, normalized

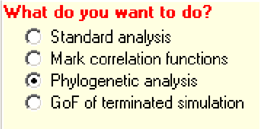


dr = 3, non-cumulative, normalized

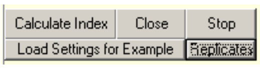


2.7.4 Show results of previous analyses: multivariate analysis

Show the results of previous multivariate analyses works in the same way as for mark correlation functions. Again, you can also change *a posteriori* the ring width and use the cumulative multivariate summary functions if appropriate. Note that some summary functions such as that from the ISAR family are already cumulative.



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation



To take advantage of these features, the original analysis must be done with ring width of 1: 1 ring widht.

Note that the *.rep files of the multivariate analysis are of the form:

“mcf_name_phy.rep”

where “name is the name you select for the results file.

2.8 Run series of analyses

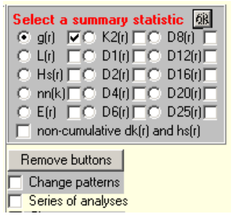
Sometimes you have to conduct many times the same analysis. *Programita* allows you to do this in an automated way. There are three different possibilities for this:

- using numbered files
- select the files to be analyzed from a list (only mode for mark correlation functions)
- for the bivariate standard analysis you can analyze all pairs of univariate patterns defined in two file lists. This is practical if you have many pairs to analyze because you need not to store all data files as in the first two cases.

Programita conducts many individual analyses, outputs results files for each analysis, and one summary file that provides an overview over all analyses.

2.8.1 Standard analysis: multiple analyses with numbered files

The first step is to conduct the analysis with one of the data files. Use for example the analysis “Book_Fig4_2b.res”. Once this is done, select the check box “**Series of analysis**”. There are 10 data file with names “test_1.dat”, “test_2.dat”, ..., “test_10.dat” to be analyzed in the same way as “Book_Fig4_2b.res”.



Select a summary statistic

● g(r) ☑ ○ K2(r) □ ○ D8(r) □
○ L(r) □ ○ D1(r) □ ○ D12(r) □
○ Hs(r) □ ○ D2(r) □ ○ D16(r) □
○ nn(k) □ ○ D4(r) □ ○ D20(r) □
○ E(r) □ ○ D6(r) □ ○ D25(r) □
□ non-cumulative dk(r) and hs(r)

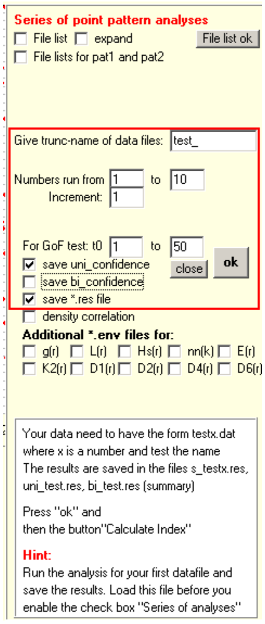
□ Change patterns
□ Series of analyses

A window opens where you need to provide the specifications of your series of analysis in the area of the red box. Your data files must all follow the name convention “name_n.dat” where name is a name you can provide in the “**Give trunk name of data files**” and n is the number of the data file. You can specify the first number and the last number and an increment. You have “test_” and n = 1,..., 10.

Programita conducts a Goodness-of-fit (GoF) test for each analysis. You can specify the distance interval over which to conduct the GoF test (here 1 – 50).

Programita can output for the analysis of each data file several results files:

- uni_confidence: the observed univariate summary function of the data (first line) and of all simulations of the null model (following lines)
- bi_confidence: same as uni_confidence, but for bivariate functions.
- *.res: this is the results file which contains all your settings. This file should be outputted.
- a *.txt summary file with name Summary_name.txt
- if you check additional summary functions at “**additional *.env files for**” *Programita* outputs the corresponding *.env files.



Series of point pattern analyses

□ File list □ expand
□ File lists for pat1 and pat2

Give trunc-name of data files: test_

Numbers run from 1 to 10
Increment: 1

For GoF test: t0 1 to 50
☑ save uni_confidence
□ save bi_confidence
☑ save *.res file
□ density correlation

Additional *.env files for:
□ g(r) □ L(r) □ Hs(r) □ nn(k) □ E(r)
□ K2(r) □ D1(r) □ D2(r) □ D4(r) □ D6(r)

Your data need to have the form testx.dat where x is a number and test the name The results are saved in the files s_testx.res, uni_test.res, bi_test.res (summary)

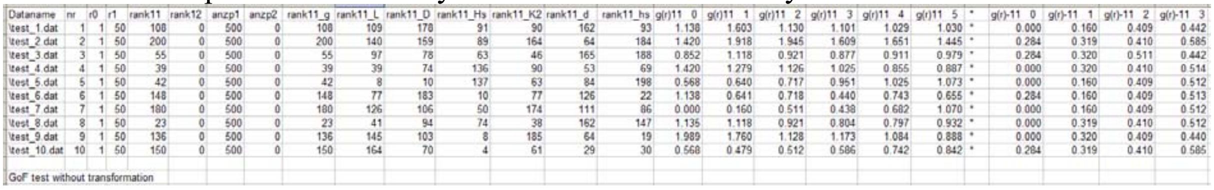
Press "ok" and then the button"Calculate Index"

Hint:
Run the analysis for your first datafile and save the results. Load this file before you enable the check box "Series of analyses"

Once all settings are specified, click the fat **ok** bottom ok, and then “**Calculate Index**” to start the series of analyses.

After termination of the simulation series, you can load the comma delimited summary file into EXCEL. The summary file has the name Summary_truncname.txt where truncname is the shared name of all data files you provided in Give trunc-name of data files: test_ :

This is an example of the summary file for an univariate analysis:



Dataname	nr	r0	r1	rank11	rank12	anzp1	anzp2	rank11_g	rank11_L	rank11_D	rank11_Hs	rank11_K2	rank11_d	rank11_hs	g(r)11 0	g(r)11 1	g(r)11 2	g(r)11 3	g(r)11 4	g(r)11 5	*	g(r)-11 0	g(r)-11 1	g(r)-11 2	g(r)-11 3
\test_1.dat	1	1	50	108	0	500	0	108	109	178	91	90	162	93	1.138	1.603	1.130	1.101	1.029	1.030	*	0.000	0.160	0.409	0.442
\test_2.dat	2	1	50	200	0	500	0	200	140	159	89	164	64	184	1.420	1.918	1.945	1.609	1.651	1.445	*	0.284	0.319	0.410	0.585
\test_3.dat	3	1	50	55	0	500	0	55	97	78	63	46	165	188	0.852	1.118	0.921	0.877	0.911	0.979	*	0.284	0.320	0.511	0.442
\test_4.dat	4	1	50	39	0	500	0	39	39	74	136	90	53	69	1.420	1.279	1.126	1.025	0.855	0.887	*	0.000	0.320	0.410	0.514
\test_5.dat	5	1	50	42	0	500	0	42	8	10	137	63	84	198	0.568	0.640	0.717	0.951	1.025	1.073	*	0.000	0.160	0.409	0.512
\test_6.dat	6	1	50	148	0	500	0	148	77	183	10	77	126	22	1.138	0.641	0.718	0.440	0.743	0.655	*	0.284	0.160	0.409	0.513
\test_7.dat	7	1	50	180	0	500	0	180	126	106	50	174	111	86	0.000	0.160	0.511	0.438	0.682	1.070	*	0.000	0.160	0.409	0.512
\test_8.dat	8	1	50	23	0	500	0	23	41	94	74	38	162	147	1.135	1.118	0.921	0.804	0.797	0.932	*	0.000	0.319	0.410	0.512
\test_9.dat	9	1	50	136	0	500	0	136	145	103	8	185	64	19	1.989	1.760	1.128	1.173	1.084	0.888	*	0.000	0.320	0.409	0.440
\test_10.dat	10	1	50	150	0	500	0	150	164	70	4	61	29	30	0.568	0.479	0.512	0.586	0.742	0.842	*	0.284	0.319	0.410	0.585
GoF test without transformation																									

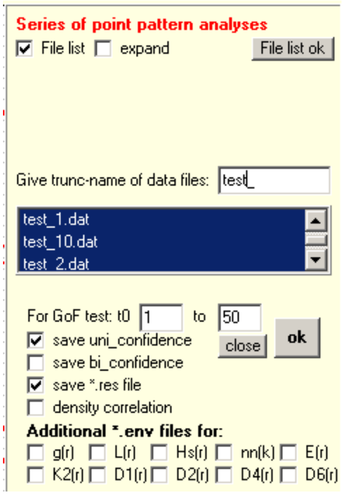
- (r0, r1) is interval of the GoF test,
- “rank 11” is the rank of the GoF test for the univariate analysis,
- “rank12” is the rank of the GoF test for the bivariate analysis,
- “anzp1” and “anzp2” give the number of points of pattern 1 and 2, respectively.
- The “rank11_g”, “rank11_L”, “rank11_D”, “rank11_Hs”, “rank11_K2” show the rank of the GoF test for the summary functions *g*(*r*), the *L*(*r*), *D*(*r*), $H_s(r)$, and *K2*(*r*), respectively.
- The following lines are observed summary functions and pointwise simulation envelopes, g11 is *g*(*r*), g-11 and g+11 are the lower and upper simulation envelopes of *g*(*r*), etc.
- The rank0, rank1, .. , rankr give the rank of the GoF test of the selected summary function at distance *r*.

In the **standard grid based mode**, you can only output the *O*(*r*) or the *L*(*r*) in the summary file, and additionally the *D*(*r*) as *.env file.

2.8.2 Multiple analyses with files selected from a list

As above, the first step is to conduct the analysis with one of the data files. Once this is done, select the check box “**Series of analysis**”.

A window opens where you need to provide the specifications of your series of analysis. To select files from the working directory select “**File list**”. A list with files appears. To enlarge the file list click “**expand**”.



Series of point pattern analyses

☑ File list ☐ expand

File list ok

Give trunc-name of data files: test_

test_1.dat
test_10.dat
test_2.dat

For GoF test: t0 1 to 50

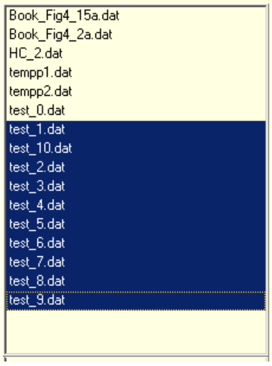
☑ save uni_confidence
☐ save bi_confidence
☑ save *.res file
☐ density correlation

close

ok

Additional *.env files for:

☐ g(r) ☐ L(r) ☐ Hs(r) ☐ nn(k) ☐ E(r)
☐ K2(r) ☐ D1(r) ☐ D2(r) ☐ D4(r) ☐ D6(r)



Book_Fig4_15a.dat
Book_Fig4_2a.dat
HC_2.dat
tempp1.dat
tempp2.dat
test_0.dat
test_1.dat
test_10.dat
test_2.dat
test_3.dat
test_4.dat
test_5.dat
test_6.dat
test_7.dat
test_8.dat
test_9.dat

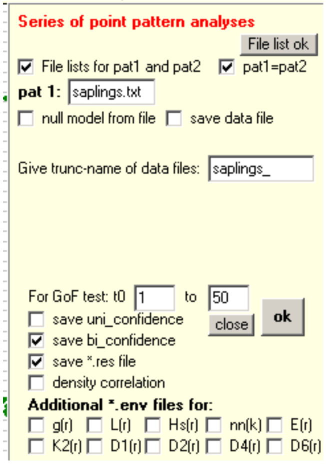
Now you can select all files you want to analyze. The trunk-name is now only used to name the summary output file.

If your data are selected, click “**File list ok**” to confirm your selection.

Once all settings are specified, click the fat **ok** bottom ok, and then “**Calculate Index**” to start the series of analyses.

2.8.3 Multiple bivariate standard analyses with all pairs of files from two file lists

As above, the first step is to conduct one analysis with a bivariate data file (use the example file “sapling2vs1.res”. Once this is done, select the check box “**Series of analyses**”.



Series of point pattern analyses

File list ok

☑ File lists for pat1 and pat2 ☑ pat1=pat2

pat 1: saplings.txt

□ null model from file □ save data file

Give trunc-name of data files: saplings_

For GoF test: t0 1 to 50

□ save uni_confidence

☑ save bi_confidence

☑ save *.res file

□ density correlation

close ok

Additional *.env files for:

□ g(r) □ L(r) □ Hs(r) □ nn(k) □ E(r)

□ K2(r) □ D1(r) □ D2(r) □ D4(r) □ D6(r)

A window opens where you need to provide the specifications of your series of analyses. To select files for pairwise analyses select “**File list for pat1 and pat2**”.

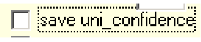
If pattern 1 and 2 should be selected from the same list (e.g., bivariate analyses of recruits of different species) select “**pat1=pat2**”. This is necessary to omit that the same file is selected as pattern 1 and pattern 2.

Insert the name of the file list for pat 1 (and if appropriate, for pat 2). The file list is an *.txt ASCII file with the names of the files to be analyzed (but without the *.dat extension).

this is an example of a file list (saplings.txt):

```
saplings1
saplings2
saplings3
```

The trunk-name is now only used to name the summary output file. For bivariate analyses, you need usually no output for the univariate analysis, therefore de-select the univariate output with

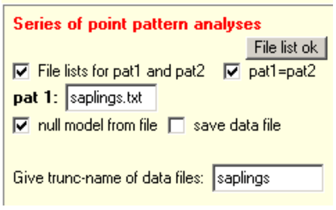


□ save uni_confidence

If your data are selected, click “File list ok” to confirm your selection.

Once all settings are specified, click the fat **ok** bottom ok, and then “**Calculate Index**” to start the series of analyses.

You can use this analysis series also for cases where you previously saved the null model for the different patterns listed in the file lists, for example generated with **pattern reconstruction**.



Series of point pattern analyses

File list ok

☑ File lists for pat1 and pat2 ☑ pat1=pat2

pat 1: saplings.txt

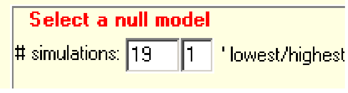
☑ null model from file □ save data file

Give trunc-name of data files: saplings

In this case you need to click “null model from file”. The null model patterns corresponding to your data files must follow the name conventions:

data file:	`name.dat`
null model file:	`rec_name_n.dat`

where `name` is the data file (e.g., Saplings1 in the example from the file list above) and `n` the number that should run from 1 to the number of # simulations of the null model specified in the window “**Select a null model**”.

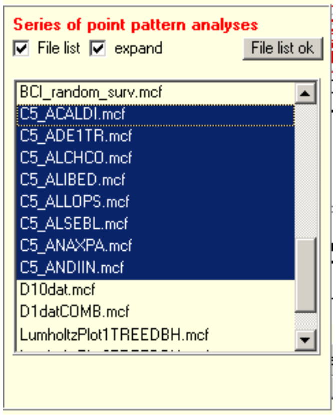


Select a null model

simulations: 19 1 ' lowest/highest

2.8.4 Multiple analyses with mark correlation functions

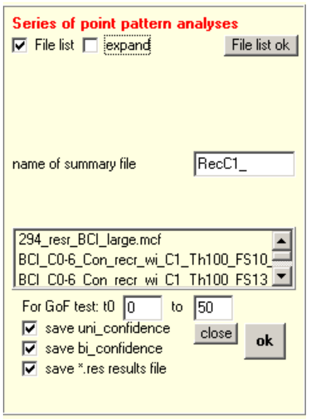
Running series of analyses works for mark correlation functions in the same way as for the standard analysis but only with the list option. The first step is to conduct the analysis with one of the data files. Once this is done, select the check box “**Series of analyses**”. A window opens where you need to select “**File list**”. A list with files appears. To enlarge the file list click “expand”.


Series of point pattern analyses
☑ File list ☑ expand

BCI_random_surv.mcf
C5_ACALDI.mcf
C5_ADE1TR.mcf
C5_ALCHCO.mcf
C5_ALIBED.mcf
C5_ALLOPS.mcf
C5_ALSEBL.mcf
C5_ANAXPA.mcf
C5_ANDIIN.mcf
D10dat.mcf
D1datCOMB.mcf
LumholtzPlot1TREEDBH.mcf

Now you can select all files you want to analyze. If your data are selected, click “**File list ok**” to confirm your selection. Provide also the name of the file that contains a summary of the results of the series of analyses.

To get the *.env files for the GoF test of individual analyses click “**save uni_confidence**” and “save **bi_confidence**” if appropriate. Once all settings are specified, click the fat **ok** bottom ok, and then “**Calculate Index**” to start the series of analyses.

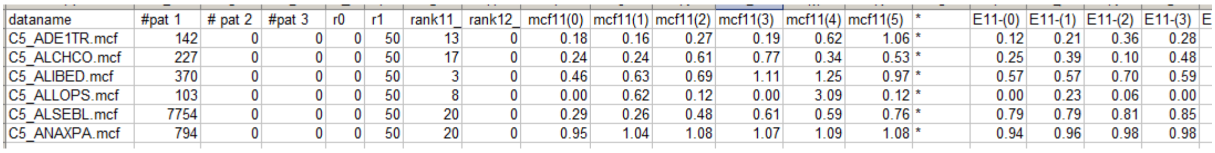

Series of point pattern analyses
☑ File list □ expand

name of summary file RecC1_

294_resr_BCI_large.mcf
BCI_C0-6_Con_recr_wi_C1_Th100_FS10_
BCI_C0-6_Con_recr_wi_C1_Th100_FS13

For GoF test: t0 0 to 50
☑ save uni_confidence
☑ save bi_confidence
☑ save *.res results file

After termination of the simulation series, you can load a comma delimited summary file (name convention “**Summary_mcf_name.dat**”) into EXCEL. This is an example of the summary file for an univariate analysis:

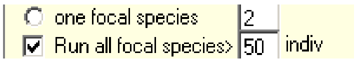


dataname	#pat 1	# pat 2	#pat 3	r0	r1	rank11_	rank12_	mcf11(0)	mcf11(1)	mcf11(2)	mcf11(3)	mcf11(4)	mcf11(5)	*	E11-(0)	E11-(1)	E11-(2)	E11-(3)	E
C5_ADE1TR.mcf	142	0	0	0	50	13	0	0.18	0.16	0.27	0.19	0.62	1.06	*	0.12	0.21	0.36	0.28	
C5_ALCHCO.mcf	227	0	0	0	50	17	0	0.24	0.24	0.61	0.77	0.34	0.53	*	0.25	0.39	0.10	0.48	
C5_ALIBED.mcf	370	0	0	0	50	3	0	0.46	0.63	0.69	1.11	1.25	0.97	*	0.57	0.57	0.70	0.59	
C5_ALLOPS.mcf	103	0	0	0	50	8	0	0.00	0.62	0.12	0.00	3.09	0.12	*	0.00	0.23	0.06	0.00	
C5_ALSEBL.mcf	7754	0	0	0	50	20	0	0.29	0.26	0.48	0.61	0.59	0.76	*	0.79	0.79	0.81	0.85	
C5_ANAXPA.mcf	794	0	0	0	50	20	0	0.95	1.04	1.08	1.07	1.09	1.08	*	0.94	0.96	0.98	0.98	

- #pat 1, #pat 2, and #pat 3 gives the number of points of type 1, 2 or 3 points, respectively,
- (r0, r1) is interval of the GoF test,
- rank 11 is the rank of the GoF test for the univariate analysis,
- rank12 is the rank of the GoF test for the bivariate analysis.
- The following lines are the observed summary functions (using the mark correlation function you selected in the example analysis) and simulation envelopes (indicated by mvf11, E-11, E+11), and mcf11_exp is the expectation of the mark correlation function.

2.8.5 Multiple analyses for multivariate analysis using a dissimilarity matrix

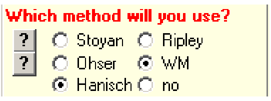
I did not implement a Series of analyses option for this data type because it will be in most cases a community level analysis. However, there is a possibility to conduct series of “individual” analyses where the individuals of a given species are selected as focal points and the individuals of the entire community are used as second points at distance *r* away. To use this option enable the checkbox “**Run all focal species**” in the window specifying the null model:


○ one focal species 2
☑ Run all focal species> 50 indiv

2.9 Combine results from replicate analyses

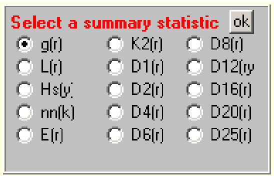
In some cases you may have maps of several replicate plots of a larger point pattern under identical conditions. In this case the resulting test statistics of the individual replicate plots can be combined into average test statistics (Diggle 2003: page 123; Illian et al. 2008: page 263; Wiegand and Moloney 2014: section 3.2). This is of particular interest if the number of points in each replicate plot is relatively low. In this case the simulation envelopes of individual analyses would become wide, but combining the data of several replicate plots into average test statistics increases the sample size and thus narrows the simulation envelopes. When considering different species as replicates, average test statistics are also an effective way of summarizing the results of an analysis on the community level. Average test statistics based on replicates can also be used to implement specific null models that would otherwise require very specific software. Section 3.2.1 of Wiegand and Moloney (2014) provides details on the aggregation formulas for different summary functions and section 3.2.2 several examples.

The default estimators of the *Programita* standard mode (and the grid-based mode) use the WM estimators for the pair correlation and the *K*-function based on the quantities $\lambda g(r)$ and $\lambda K(r)$. The corresponding aggregation formulas for the WM estimator are provided in equations 3.114 and 3.117 in Wiegand and Moloney (2014). For the other estimators available in the standard mode (Stoyan, Ripley, and Ohser) the aggregation formula for $g(r)$ and $K(r)$ is based on the abundance weighted mean of the $g(r)$ and $K(r)$ for the individual plots where the weight is the abundance of the focal pattern 1 [because it combines $\lambda g(r)$ and $\lambda K(r)$].



Which method will you use?
? ○ Stoyan ○ Ripley
? ○ Ohser ● WM
● Hanisch ○ no

If you select other summary functions than $g(r)$ or $L(r)$, the aggregation formula used by *Programita* is also the abundance weighted mean of the summary functions of the individual replicates where the weight is the abundance of the focal pattern 1. Note that it makes little sense to apply the aggregation formula to the K2 function, here you should first estimate the pair correlation function and then estimate the derivative.



Select a summary statistic ok
● g(r) ○ K2(r) ○ D8(r)
○ L(r) ○ D1(r) ○ D12(r
○ Hs(y ○ D2(r) ○ D16(r)
○ nn(k) ○ D4(r) ○ D20(r)
○ E(r) ○ D6(r) ○ D25(r)

2.10 Settings and estimators of the summary functions

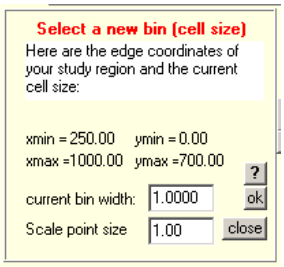
2.10.1 Bins and distances of non-cumulative summary functions

Depending on the summary functions used, *Programita* uses conventions to define distances and the distance bins. In general, second-order summary functions based on product densities (e.g., pair correlation functions, K2 function, mark connection functions, and mark correlation functions) characterize properties of the spatial pattern using pairs of points i and j that are located approximately **at distance *r*** of each other.

Programita uses simple **box kernels** to define “at distance *r*”. This kernel function introduces a small “tolerance” interval ($r - dr/2$, $r + dr/2$) for the distance r (called the bandwidth $dr/2$) within which two points are regarded as being located distance r apart. In this way, the kernel function defines rings with width dr and radius r around the focal point i and any point j falling within the ring causes the kernel function to evaluate to a positive value (less than or equal to one), otherwise it returns zero.

Programita uses a default bin width of 1 (i.e., it uses the units of the data), however, when you select (in the standard mode) a data file the window Select a new bin (cell size) opens and allows you to change the unit of the data. For example, if you select 5, all analyses are conducted with bins of 5m.

In the mark correlation modes this information can be selected in the windows Mark correlation function or Multivariate analysis.



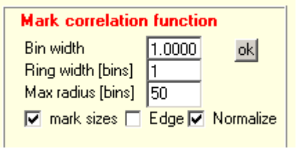
Select a new bin (cell size)

Here are the edge coordinates of your study region and the current cell size:

xmin = 250.00 ymin = 0.00
xmax =1000.00 ymax =700.00

current bin width: 1.0000

Scale point size 1.00



Mark correlation function

Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50
☑ mark sizes ☐ Edge ☑ Normalize

Programita initially locates each point pair i and j within the following distance bins:
0.5: [0 - 1)
1.5: [1 - 2)
2.5: [2 - 3)
…
where the “[“ and “)” indicate that the left but not right border belongs to the interval. If you select a bin width different from 1 (e.g., 5), the bins and intervals have to be scaled to yield bins in the original data units: 2.5: [0 - 5), 7.5: [5 - 10), 2.5: [20 - 15), …

This binning corresponds to a ring width $dr = 1$. However, you can select a larger ring width dr of 3, 5, 7, … bins. In this case the binning uses not only the central bin, but also the neighboring bins and the binning yields e.g., for $dr = 3$ (in units of bins):
0.5: [0 - 2), 1.5: [0 - 3), 2.5: [1 - 4), … Note that the first bins may be smaller than dr.

The *.res output file provides the distance bin in units of the data.

The following summary functions in the standard analysis mode are scaled in this way: $g(r)$, $E(r)$, $K2(r)$, and mark connection and the non-cumulative mark correlation functions.

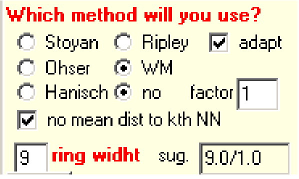
2.10.2 Bins and distances of cumulative summary functions

Estimation of cumulative summary functions is easier. In this case no kernel function is required and the maximal distance *r* means simply distances from zero up to *r*, i.e., interval [0, r).

If you select a bin width different from 1 (e.g., 5), the bins have to be scaled to yield bins in the original data units: [0 - 5), [0 - 10), [0 - 20), … The *.res output file then provides the distance bin *r* in units of the data.

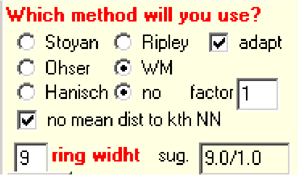
This rule applies for the cumulative summary functions such as *L* functions $L(r)$, nearest neighbor distribution functions $D^k(r)$, the spherical contact distribution $H_s(r)$, and cumulative mark correlation functions (available when using combine replicates).

2.10.3 Estimators in standard analysis

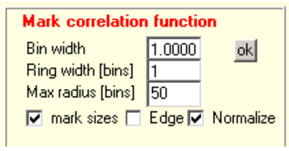

Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1
☑ no mean dist to kth NN
9 ring widht sug. 9.0/1.0

For the estimation of second order summary functions in the standard analysis mode you can select among four methods of edge correction which are detailed in Wiegand and Moloney (2014: equations 3.24, 3.37, 3.54, and 3.61). The Ripley and Stoyan method corrects for each point pair individually using a weight w_{ij}. The Ripley weight divides the circumference of a full disk centered on point *i* with radius *r* and passing through point *j* by the circumference of the disk lying inside the sample domain and the Stoyan edge correction uses the translation method. The Ohser and WM edge correction correct not each point pair individually, but use for each distance *r* a factor that corrects globally.

For the nearest neighbor distribution functions $D^k(r)$ you can select "no edge correction" and the so-called Hanisch edge correction (see sections 3.1.3.1, 3.1.3.1, and 3.1.4.5) in Wiegand and Moloney (2014). The spherical contact distribution $H_s(r)$ is estimated as bivariate $D_{12}(r)$ (where a regular grid of 40 × 40 test points is pattern 1 and the univariate pattern is pattern 2) using the Hanisch edge correction. You can increase the number of test points by factor *f* in each direction (but this may slow down the estimation).


Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1
☑ no mean dist to kth NN
9 ring widht sug. 9.0/1.0

The mean distance to the kth neighbor $nn(k)$ is estimated without edge correction. Because the estimation of $nn(k)$ slows down *Programita*, you can reduce the maximal neighborhood rank *k* to the maximal *k* used for the $D^k(r)$ by enabling "no mean dist to kth NN".


Mark correlation function
Bin width 1.0000 ok
Ring width [bins] 1
Max radius [bins] 50
☑ mark sizes □ Edge ☑ Normalize

For mark correlation functions you can use the Stoyan translation edge correction when clicking the check box "edge", but there will be almost no effect since the edge correction terms cancel in the ratio estimator of mark correlation functions.

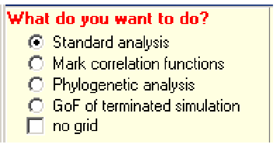
3 Univariate analysis in the standard analysis mode

The standard analysis mode allows for analysis of

- univariate patterns (data type 1)
- bivariate patterns (data type 2)
- qualitatively marked patterns (i.e., random labeling analysis; data type 4).

In the following I explain the different settings, null models, and point process models for univariate analysis in detail. Most examples are taken from Wiegand and Moloney (2014).

The standard analysis mode works in most cases exactly the same way as the grid-based standard mode. The only difference is to enable or disable the checkbox "no grid" in the window What do you want to do?



What do you want to do?

● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
□ no grid

I therefore report here only cases where differences occur. For example, in some cases the procedures for the standard mode are not yet implemented and you have to use instead the grid-based mode for particular analyses.

3.1 Getting started

3.1.1 Data preparation

Univariate analysis deals with a data type that comprises only the coordinates of a given point pattern. There is only one type of points and no mark considered. The univariate data type is the most analyzed data type in point pattern analysis.

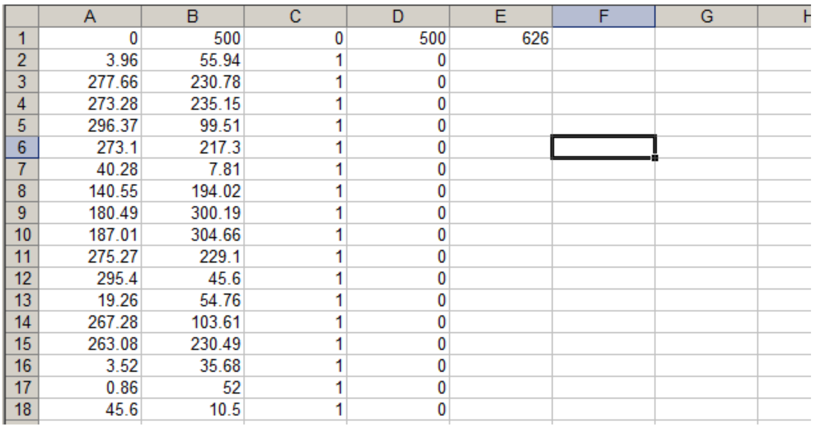
The data files for univariate standard analysis must be an ASCII file with the *.dat extension and the following format (the example shows the first lines of the file Book_Fig4_15a.dat):

```
0   500 0   500 626
  3.96    55.94    1    0
277.66   230.78    1    0
273.28   235.15    1    0
296.37    99.51    1    0
273.10   217.30    1    0
 40.28     7.81    1    0
140.55   194.02    1    0
180.49   300.19    1    0
187.01   304.66    1    0
275.27   229.10    1    0
…
```

where the first line gives the size of the observation window (500 × 500 units in the example) and the number of points in the pattern (= number of lines following the header). The first two columns are the coordinates, an entry "1" in the third column indicates that the point is of pattern 1 (i.e., a type 1 focal point) and an entry "1" in the fourth column indicates that the point is of pattern 2 (i.e., a type 2 point). The value of the third and the forth columns must be for the standard analysis mode "0 1" or "1   0", no larger numbers or "1 1" are allowed.

The data file must be a space or tab delimited ASCII file with the *.dat extension. If you use Excel, there is a simple, but obviously generally unknown, way of saving files of a given type with a given extension:

1. Prepare the data file in Excel following the instructions above
2. Then save as a tab delimited text file, but write "name.dat" for the name (usually you would only write name and end up with a file named name.txt. The quotation marks are important because they force Excel to save the comma delimited file under the name name.dat.



	A	B	C	D	E	F	G	H
1	0	500	0	500	626			
2	3.96	55.94	1	0				
3	277.66	230.78	1	0				
4	273.28	235.15	1	0				
5	296.37	99.51	1	0				
6	273.1	217.3	1	0				
7	40.28	7.81	1	0				
8	140.55	194.02	1	0				
9	180.49	300.19	1	0				
10	187.01	304.66	1	0				
11	275.27	229.1	1	0				
12	295.4	45.6	1	0				
13	19.26	54.76	1	0				
14	267.28	103.61	1	0				
15	263.08	230.49	1	0				
16	3.52	35.68	1	0				
17	0.86	52	1	0				
18	45.6	10.5	1	0				

3.1.2 Steps of analysis in standard mode and example

Programita estimates for data files of this type several summary functions based on estimators detailed in Illian et al. (2008) and Chapter 3 of Wiegand and Moloney (2014). The window Which method will you use allows you to specify the estimators.

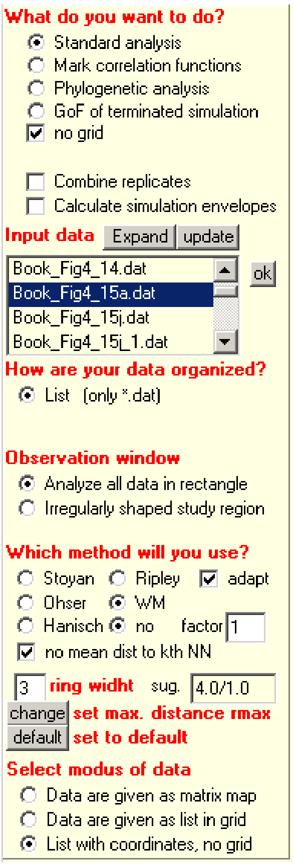
The standard analysis mode can be accessed with the following sequence of actions:

1. Highlight a data file “Book_Fig4_15a.dat” in Input data and click the small “ok” button.
2. The window Select a new cell size opens and allows you to provide a bin for your analysis given in units of your data. For example, if your data are in meter units and your observation window is 500 × 500m in size, an appropriate bin would be 1m. Press “**ok**” to confirm selection of the bin.

3. After selection of the bin ***Programita* suggests a ring width *dr*** for the estimation of the pair correlation function $g(r)$ based on equation 4.3.43 in Illian et al. (2008) [$dr = 0.2/\lambda^{0.5}$]. This equation provides a rough starting point for deciding on the ring width. In the example file “Book_Fig4_15a.dat” with 626 points within a 500 × 500m observation window and a bin of 1m this yields a ring width of $dr = 4.0$ for pattern 1. Because the pattern is strongly clustered and dr can only have values of $dr = 1, 3, 5, \ldots$ select $dr = 3$.

4. The estimators of the pair correlation function implemented in the standard mode of *Programita* use a **default ring width of one bin** to obtain non-overlapping concentric rings. For reasons of computational efficiency you can then select only ring widths adding one bin in each direction, i.e., ring widths of 1, 3, 5, 7, … bins. You can change the ring width at the menu “Which method will you use”. In the example you may use a ring width of 3.

5. Selecting the option “**no grid**” opens also a small window where you can select the desired rank k of the distribution functions $D^k(r)$ of the distances to the kth neighbor. Default is $k = 1, 2, 4, 6, 8, 12, 16, 20$, and 25. You can thus select the rank k of nine different functions $D^k(r)$. To confirm press the small **ok** button.



What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

□ Combine replicates
□ Calculate simulation envelopes

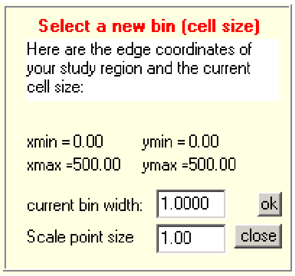
Input data Expand update
Book_Fig4_14.dat
Book_Fig4_15a.dat
Book_Fig4_15j.dat
Book_Fig4_15j_1.dat
ok

How are your data organized?
● List (only *.dat)

Observation window
● Analyze all data in rectangle
○ Irregularly shaped study region

Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1
☑ no mean dist to kth NN
3 ring widht sug. 4.0/1.0
change set max. distance rmax
default set to default

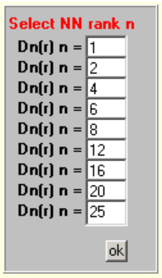
Select modus of data
○ Data are given as matrix map
○ Data are given as list in grid
● List with coordinates, no grid



Select a new bin (cell size)
Here are the edge coordinates of your study region and the current cell size:

xmin = 0.00 ymin = 0.00
xmax =500.00 ymax =500.00

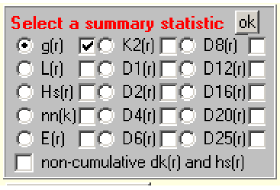
current bin width: 1.0000 ok
Scale point size 1.00 close



Select NN rank n
Dn(r) n = 1
Dn(r) n = 2
Dn(r) n = 4
Dn(r) n = 6
Dn(r) n = 8
Dn(r) n = 12
Dn(r) n = 16
Dn(r) n = 20
Dn(r) n = 25
ok

6. Press button “**Calculate Index**” and *Programita* estimates a variety of summary functions of the univariate data:
   - $g(r)$: pair correlation function
   - $L(r)$: L-function,
   - $H_s(r)$: the spherical contact distribution
   - $nn(k)$ the mean distance to the kth neighbor
   - $E(r)$ the probability that a point has no neighbor at distance within distances (r - $dr/2$, $r + dr/2$)
   - $K2(r)$ the $K2$ function
   - $D^k(r)$, the kth nn distribution functions, here with k = 1, 2, 4, 6, 8, 12, 16, 20, and 25
   - If you enable the check box “non-cumulative dk(r) and hs(r)” you can view the non-cumulative counterparts of the cumulative $D^k(r)$ and $H_s(r)$.

   To view the different summary functions select the respective radio button and then the small **ok** button.

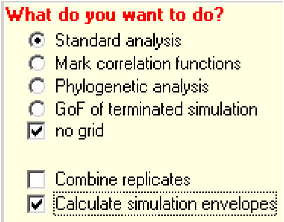
   

   Select a summary statistic [ok]

   ● g(r) ☑ ○ K2(r) □ ○ D8(r) □
   ○ L(r) □ ○ D1(r) □ ○ D12(r) □
   ○ Hs(r) □ ○ D2(r) □ ○ D16(r) □
   ○ nn(k) □ ○ D4(r) □ ○ D20(r) □
   ○ E(r) □ ○ D6(r) □ ○ D25(r) □
   □ non-cumulative dk(r) and hs(r)

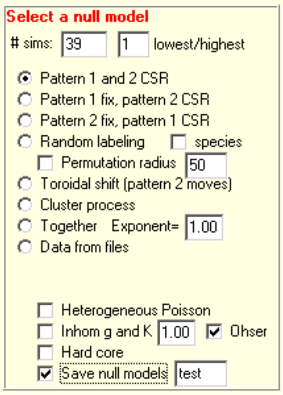
   After one analysis *Programita* saves the results of all 15 univariate (and if appropriate all 15 bivariate) summary functions into the temporary files SumStat1.env, SumStat2.env, etc. and allows you to view all results without conducting new analysis.

7. The next step is to select a **null model or point process** model implemented in *Programita*. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface. A window will open that allows you to select a null model. In the example, we select “**Pattern 1 and 2 CSR**”. In this case both patterns are independently distributed following a homogeneous Poisson processes (or Complete Spatial Randomness CSR). If the data set is univariate only the first pattern is randomized following CSR.

   What do you want to do?
   ● Standard analysis
   ○ Mark correlation functions
   ○ Phylogenetic analysis
   ○ GoF of terminated simulation
   ☑ no grid
   □ Combine replicates
   ☑ Calculate simulation envelopes

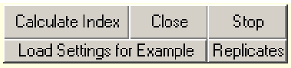
   Here you can specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).

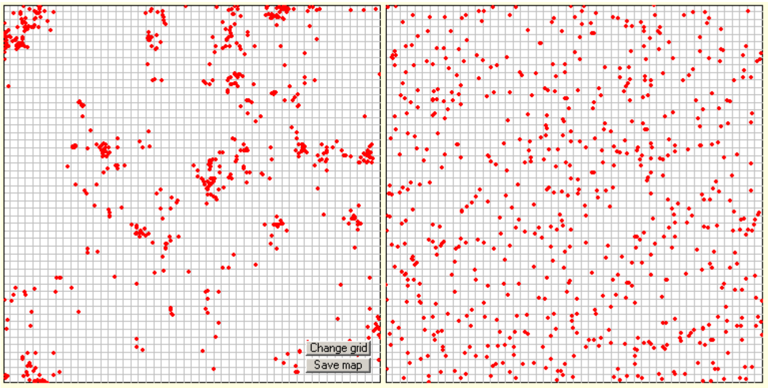
   Select a null model
   # sims: 39 1 lowest/highest
   ● Pattern 1 and 2 CSR
   ○ Pattern 1 fix, pattern 2 CSR
   ○ Pattern 2 fix, pattern 1 CSR
   ○ Random labeling □ species
   □ Permutation radius 50
   ○ Toroidal shift (pattern 2 moves)
   ○ Cluster process
   ○ Together Exponent= 1.00
   ○ Data from files
   □ Heterogeneous Poisson
   □ Inhom g and K 1.00 ☑ Ohser
   □ Hard core
   ☑ Save null models test

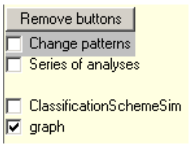
   The radio buttons in the menu “Select a null model” are the different basic options for null models or point process models whereas the check boxes are mostly additional options to specify the null model. The checkbox “**Save null models**” allows you to save the patterns generated by the null model as “name_n.dat”

   If all settings are specified, press “**Calculate Index**” and *Programita* conducts the simulations of the null model.

8. *Programita* shows the original point pattern being analyzed (left or top plot), and patterns of the Monte Carlo simulations of the null model (on the right or bottom) used for constructing the pointwise simulation envelopes and the GoF and global envelope tests.

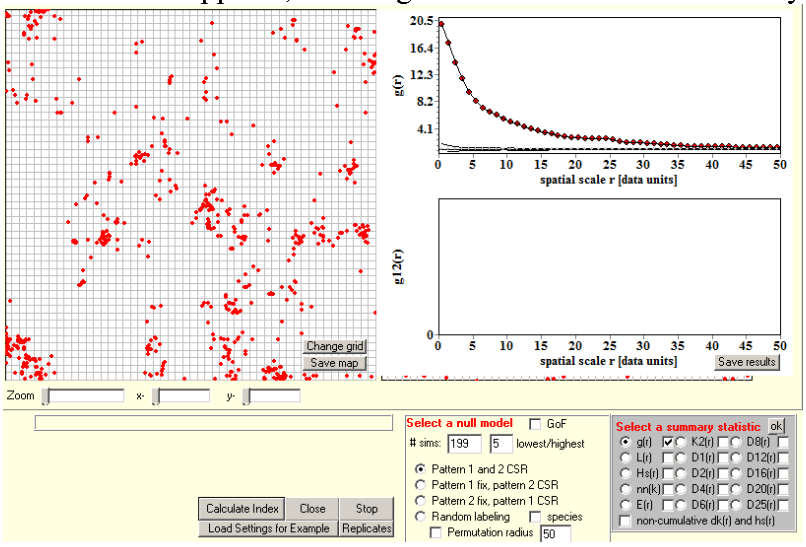




□ Change patterns
□ Series of analyses
□ ClassificationSchemeSim
☑ graph

The simulation is quicker if *Programita* does not show the plots of all simulated data. You can not show the graphs by disabling the checkbox "**graph**" at the bottom right.

After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears:



Select a null model □ GoF
# sims: 199 5 lowest/highest
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50

Select a summary statistic
● g(r) ☑ ○ K2(r) □ ○ D8(r) □
○ L(r) □ ○ D1(r) □ ○ D12(r) □
○ Hs(r) □ ○ D2(r) □ ○ D16(r) □
○ nn(k) □ ○ D4(r) □ ○ D20(r) □
○ E(r) □ ○ D6(r) □ ○ D25(r) □
□ non-cumulative dk(r) and hs(r)

The top (or left) figure shows generally the results of the univariate analysis and the bottom (right) figure shows the results of the bivariate analysis. (Exceptions are multivariate analysis using a dissimilarity matrix under the random labeling and trivariate random labeling mode). The data file in the example was univariate, therefore no figure appears for the bivariate results.

9. To save **the results of the analysis for a particular summary functions** press the button Save results in the result graph for the bivariate analysis. *Programita* then generates a *.res file [e.g., "**g(r)_name.res**" for the pair correlation function where "name" is a name] with the summary of the results and the settings of the analysis, and a *.env file with the detailed results of the summary function for the data and the simulations of the null model. The *.env file can be used for the GoF test.

3.2 Methods for univariate standard analysis

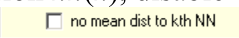
The following examples present step-by-step instructions for the most impotent univariate analyses. If analyses of Wiegand and Moloney (2014) are repeated, I refer to them using the figure number in the book, e.g., Book _Fig4_2 refers to an analysis shown in Figure 4.2 and the corresponding data file is named Book_Fig_4_2a.dat. Other analyses are named after the null model used.

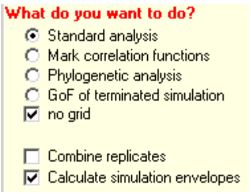
3.2.1 Homogeneous Poisson (CSR)

The homogenous Poisson process is characterized by two fundamental properties. First, the intensity λ of the process (i.e., the mean point density in a unit area) is a constant and therefore, the number of points in a study plot of area A follows a Poisson distribution with an expected mean of λA. Second, the points are independently distributed, which means that there is no interaction between the points of the pattern determining their locations.

Although CSR appears in many cases overly simple, it is the basic building block for more complex null models. As example, step-by-step instructions for the analysis of Figure 4.2 in Wiegand and Moloney (2014) are provided below.

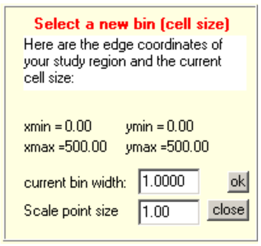
1. Execute *Programita*.
2. Highlight data file Book_Fig4_2a.dat you want to analyze in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 9 in the menu “Which method will you use”
5. Click button “**change**” below to set maximal distance *r* to be analyzed. Insert 100 in small box that opens and then the small **ok** button.
6. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
7. Press button “**Calculate Index**”
8. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
9. Select “**Pattern 1 and 2 CSR**” in the window “Select a null model”.
10. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
11. To view a large range of neighborhood ranks *k* in the mean distance to the kth neighbor summary function $nn(k)$, disable the option “**no mean distance to kth NN**” in the menu Which method will you use. This requires estimation of a larger nearest neighbor matrix which slows down the estimation. If the check box is enabled (default), the maximal *k* value is the maximal k of the $D^k(r)$’s estimated.


□ no mean dist to kth NN


What do you want to do?

● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

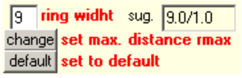
□ Combine replicates
☑ Calculate simulation envelopes

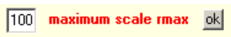

Select a new bin (cell size)

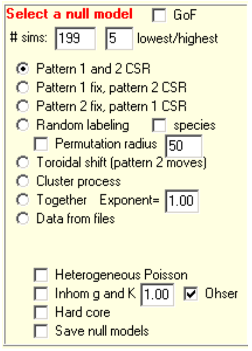
Here are the edge coordinates of your study region and the current cell size:

xmin = 0.00 ymin = 0.00
xmax =500.00 ymax =500.00

current bin width: 1.0000 ok
Scale point size 1.00 close

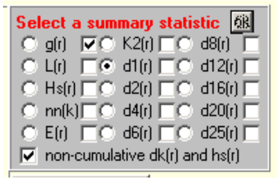

9 ring widht sug. 9.0/1.0
change set max. distance rmax
default set to default


100 maximum scale rmax ok


Select a null model □ GoF
# sims: 199 5 lowest/highest

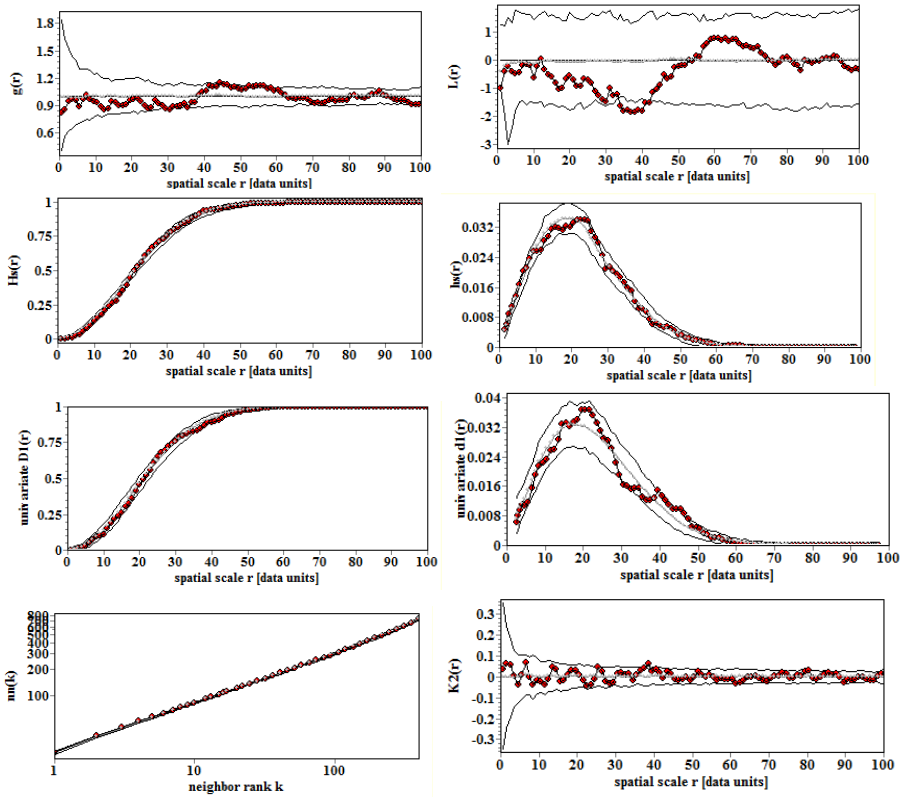
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models



Select a summary statistic
○ g(r) ☑ ○ K2(r) □ ○ d8(r) □
○ L(r) □ ● d1(r) □ ○ d12(r) □
○ Hs(r) □ ○ d2(r) □ ○ d16(r) □
○ nn(k) □ ○ d4(r) □ ○ d20(r) □
○ E(r) □ ○ d6(r) □ ○ d25(r) □
☑ non-cumulative dk(r) and hs(r)

12. If all settings are specified, press the button “**Calculate Index**” and *Programita* conducts the simulations of the null model.
13. In window “Select a summary function” you can view the results of the analysis for the different summary functions:



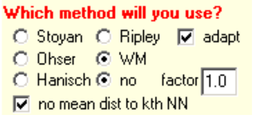
15. To save **the results of the analysis for a particular summary functions** press the button Save results that appears in the result graph for the bivariate analysis and provide the name (e.g., Book_Fig.4_2b). By enabling the small checkboxes beside the summary function you can save all selected summary functions at the same time.

 Here is an example for the *.res results file for the spherical contact distribution $H_s(r)$:

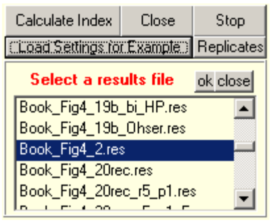
```
NN distance rr      Hs11(r)  E11-         E11+         Expect
       0.00 rr   0.0000000   0.0000000    0.0000000    0.0000000
       1.00 rr   0.0006555   0.0000000    0.0039425    0.0014869
       2.00 rr   0.0052574   0.0026458    0.0105168    0.0059499
       3.00 rr   0.0145009   0.0079239    0.0205985    0.0136494
       4.00 rr   0.0251006   0.0159655    0.0310984    0.0238473
       5.00 rr   0.0443942   0.0286351    0.0468248    0.0373350
       6.00 rr   0.0557446   0.0419402    0.0641362    0.0529284
       7.00 rr   0.0738612   0.0587447    0.0829726    0.0716262
       8.00 rr   0.0987816   0.0800635    0.1060944    0.0929994
       9.00 rr   0.1271862   0.1028497    0.1320018    0.1163091
      10.00 rr   0.1509547   0.1279122    0.1576738    0.1421021
      11.00 rr   0.1748168   0.1564811    0.1832409    0.1692070
      12.00 rr   0.2028685   0.1845011    0.2130192    0.1983967
      13.00 rr   0.2269228   0.2135108    0.2425942    0.2291874
      14.00 rr   0.2579776   0.2437670    0.2778281    0.2607731
```

 You can load this part of the results file into a scientific graphics program to produce figures.

3.2.2 Homogeneous Poisson and estimators



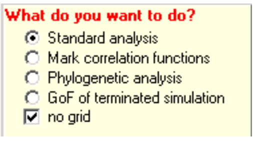
Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1.0
☑ no mean dist to kth NN



Calculate Index | Close | Stop
Load Settings for Example | Replicates

Select a results file ok close

Book_Fig4_19b_bi_HP.res
Book_Fig4_19b_Ohser.res
Book_Fig4_2.res
Book_Fig4_20rec.res
Book_Fig4_20rec_r5_p1.res



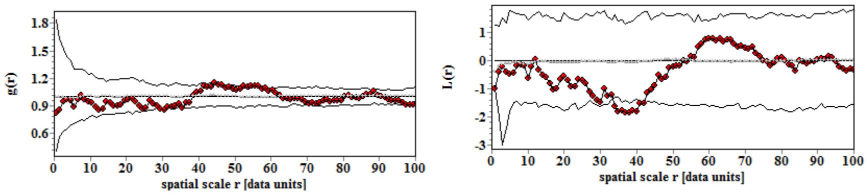
What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

In the previous example we used the WM estimator for the pair correlation function and the *L*-function. However, *Programita* allows you to use also alternative estimators presented in Illian et al. (2008) and Wiegand and Moloney (2014). The analysis Book_Fig4_2 is therefore repeated below with different estimators.

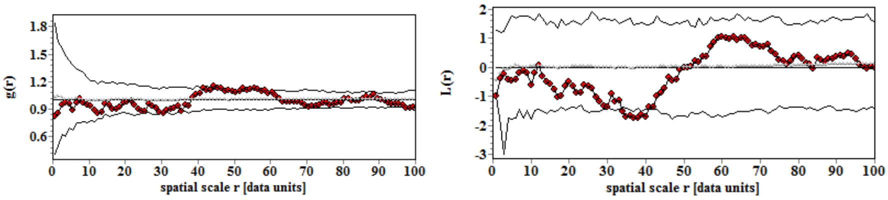
1. Click "**Load Settings for Example**", highlight file "**Book_Fig4_2.res**" and click small **ok**.
2. Select "**Stoyan**" in menu "Which method will you use".
3. Click "**Calculate Index**"
4. Repeat the same with "**Ripley**" but disable the option "adapted" to obtain the standard estimator that does not use the adapted intensity estimator proposed in Illian et al. (2008)
5. To obtain the grid-based mode deselect "**no grid**"

The differences among estimators are small for near random or random univariate patterns:

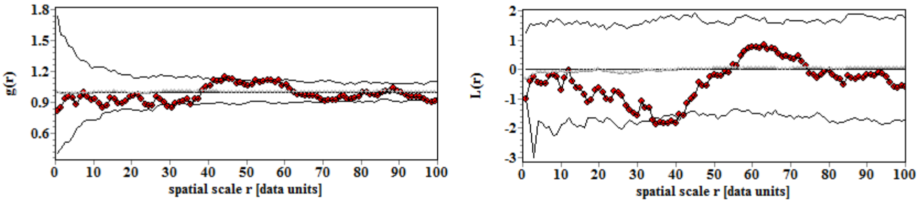
WM estimator



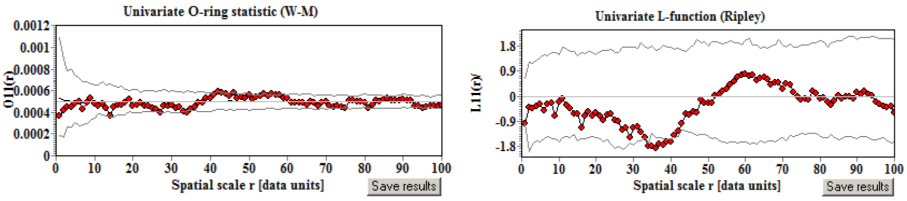
Stoyan estimator



Ripley estimator (non adapted)



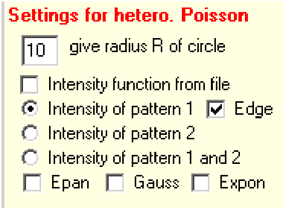
Grid-based WM estimator



3.2.3 Heterogeneous Poisson with kernel estimate

The heterogeneous Poisson process is characterized by two fundamental properties. First, in contrast to the homogeneous Poisson process the intensity $\lambda(\mathbf{x})$ of the process depends on location **x**. Second, the points are independently distributed, which means that there is no interaction between the points of the pattern determining their locations.

The heterogeneous Poisson process is completely determined by the intensity function $\lambda(\mathbf{x})$ and therefore the estimation of the intensity function is an important ingredient of this point process model. There are basically two methods to estimate the intensity function, parametric and non-parametric methods. If you used non-parametric methods to estimate the intensity function you can read the resulting intensity file into *Programita* by checking the checkbox "Intensity function from file"



Settings for hetero. Poisson

10 give radius R of circle

□ Intensity function from file

● Intensity of pattern 1 ☑ Edge

○ Intensity of pattern 2

○ Intensity of pattern 1 and 2

□ Epan □ Gauss □ Expon

However, *Programita* allows you also to estimate the intensity function non-parametrically, directly from the data using smoothing techniques based on kernel estimators. See section 2.6.2.1 "Nonparametric Intensity Estimation" in Wiegand and Moloney (2014) for details.

Programita offers four different kernel functions

- Box kernel (neither "**Epan**" nor "**Gauss**" nor "**Expon**" checked)
- Epanechnikov kernel ("**Epan**" checked)
- Gaussian kernel ("**Gauss**" checked)
- Exponential kernel ("**Expon**" checked)

that can estimate with ("**Edge**" checked) and without edge correction ("**Edge**" not checked).

Remember that the intensity is defined basically as number of points per unit area and that an estimator of the intensity divides the number of points in a given area by the area. For a homogeneous pattern the "natural" estimator of the intensity is therefore $\lambda_n = n/A$ where n is the number of points in the observation window and A the area of the observation window. The non-parametric intensity estimators generalize this idea. Because the intensity changes along the observation window it makes sense to use smaller subareas centered at location **x** to estimate the intensity function $\lambda(\mathbf{x})$. *Programita* therefore estimates the density of points within circular moving windows $C_{(\mathbf{x})}(R)$ with radius R centered on location **x**. The moving-window estimate $\hat{\lambda}^R$ of the non-constant first-order intensity $\lambda(\mathbf{x})$ yields

$$\hat{\lambda}^R(\mathbf{x}) = \frac{\textbf{Points}[C_{(\mathbf{x})}(R)]}{\textbf{Area}[C_{(\mathbf{x})}(R)]}$$

where the operator **Points**[X] counts the points in a region X, and the operator **Area**[X] determines the area of the region X. This is a **box kernel** estimate with fixed bandwidth R (all points located within distance R of location **x** are counted equal with weight 1 and all points at larger distance have weight zero and are not counted).

As edge correction, the number of points in an incomplete circle is divided by the proportion of the area of the circle that lies within the study region. Without edge correction, it is divided by the area of the full circle. The intensity function is then normalized to have a maximal value of one, thus ranging between zero and one.

The moving window estimator $\hat{\lambda}^R(\mathbf{x})$ involves a decision on an appropriate radius *R* of the moving window (see section 2.6.2.1 "Nonparametric Intensity Estimation"). As detailed in section 4.1.2.1 "HPP: Nonparametric Intensity Estimate to Avoid Virtual Aggregation", the heterogeneous Poisson process with box kernel intensity estimate has **a simple geometric interpretation**. While CSR basically displaces a given point with random distance and angle within the observation window, the heterogeneous Poisson process with box kernel intensity displaces each point of the pattern basically within a circular neighborhood of radius *R*.

Thus, because the bandwidth *R* is the scale of smoothing, possible departure from this null model may only occur for scales $r < R$, and for small moving windows it will closely mimic the original pattern, whereas a large moving window approximates CSR.

The three other options ("**Epan**", "**Gauss**", or "**Expon**") are kernel functions that weight the points which are counted according to their distance to the focal location **x**. In case of the Epanechnikov kernel (enable **Epan**") this produces smoother intensity estimates than the box kernel (Fig. 2.20 in Wiegand and Moloney 2014). The Epanechnikov kernel is based on the following weight of a point within distance *d* of location **x**:

$$w_E(d,R) = 2(1-\frac{d^2}{R^2}) \text{ if } d \leq R \text{ and 0 otherwise.}$$

The option "**Gauss**" is mostly used as analogue to the Thomas cluster process and generates an intensity function which is the superposition of two-dimensional and symmetric Gaussian curves centered in the points of the pattern (see Fig. 4.17 in Wiegand and Moloney 2014). It is based on the weight:

$$w_G(d,R) = \frac{1}{\sqrt{2\pi}R}\exp(-\frac{1}{2}(\frac{d}{R})^2) \text{ if } d \leq 3R \text{ and 0 otherwise.}$$

The option "**Expon**" generates an intensity function based on an exponential kernel

$$w_{Ex}(d,R) = \frac{1}{\sqrt{2\pi}R}\exp(-\frac{1}{2}\frac{d}{R}) \text{ if } d \leq 7R \text{ and 0 otherwise.}$$

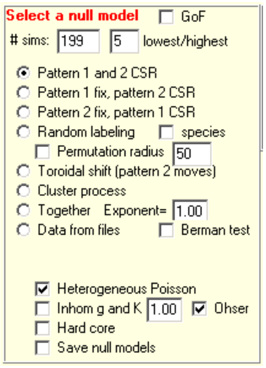
Note that the integral $\int w(r, R)\, 2\pi\, r\, dr$ over both kernel functions yields the area πR^2 of the circle. For computational reasons the Gaussian kernel is only estimated up to distances of 3R because in this case the integral yields 0.99 πR^2. Similarly, the exponential kernel is estimated up to distance 7*R* and takes longer to estimate.

The algorithm for creating a pattern under a heterogeneous Poisson process is simple: a provisional point is placed at a random cell (x, y) in the study area, but this point is only retained with probability $\hat{\lambda}^R(x,y)/\max[\hat{\lambda}^R(x,y)]$ (the function max[X] determines the maximum of a variable X). This procedure is repeated until *n* points are distributed.

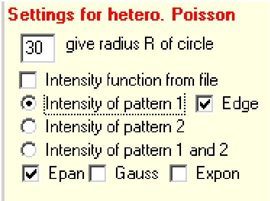
Example Book_Fig2_28h.res

The following example presents the analysis of Figure 2.28 using a heterogeneous Poisson process with non-parametric kernel estimate.

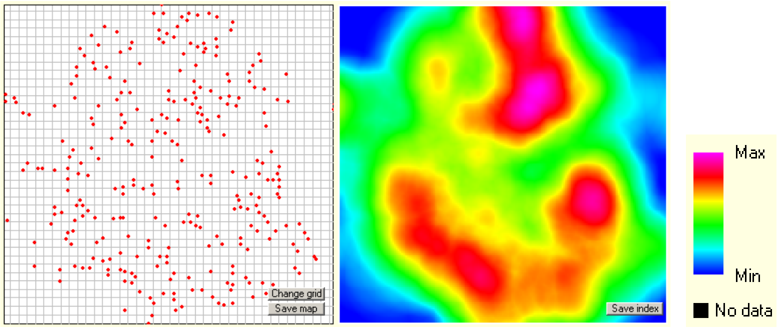
1. Execute *Programita*.
2. Highlight data file Book_Fig2_26.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Pattern 1 and 2 CSR**" in the window "Select a null model".
9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
10. Click checkbox "**Heterogeneous Poisson**"
11. Go to window "Settings for hetero. Poisson" on the left and insert the bandwidth R (30m in the example), enable "**Epan**" for the Epanechnikov kernel and select "**Intensity of pattern 1**" (because your data are univariate). Edge correction "**Edge**" is enabled by default. Click "**Calculate Index**" and *Programita* estimates the intensity function and shows the pattern and the corresponding intensity function.



Select a null model □ GoF
# sims: 199 5 lowest/highest
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test
☑ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

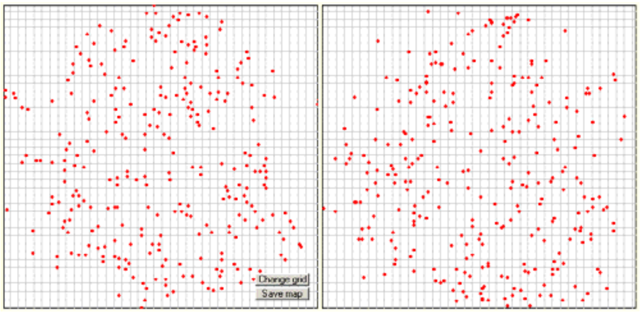


Settings for hetero. Poisson
30 give radius R of circle
□ Intensity function from file
● Intensity of pattern 1 ☑ Edge
○ Intensity of pattern 2
○ Intensity of pattern 1 and 2
☑ Epan □ Gauss □ Expon

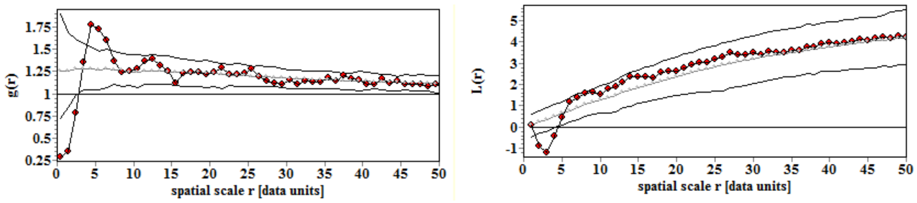


Click OK at the message box to save the intensity file. The file is saved with name int_E_Book_Fig2_26_R1_30.int where the "int_E" indicates Epanechnikov kernel, Fig2_26.dat was the data file, "_R1_30" means that the intensity was estimated with pattern 1 and bandwidth 30.

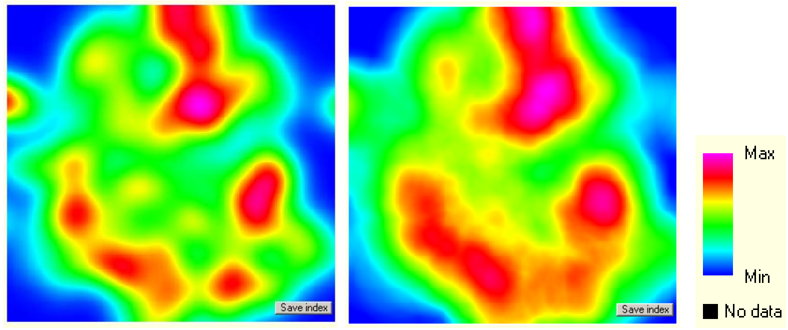
Now *Programita* conducts the analysis. You can observe during the simulations that the null model distributes the points with probability proportionally to the intensity. Here an example:



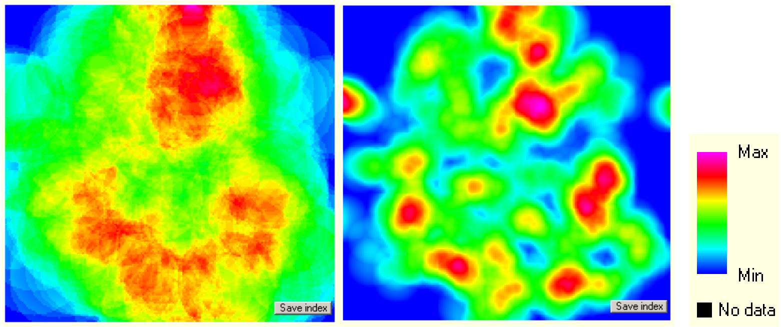
12. The result resembles that in Figure 2.28 h, i well:



13. This is the analogous Gaussian kernel (left) with R = 10m in comparison with the Epanechnikov kernel with R = 30m (right):



In contrast, the estimate using the box kernel with bandwidth R = 30m looks quite rugged (left) and an Epanechnikov kernel with R = 15m seems to conserve too much detail (right):

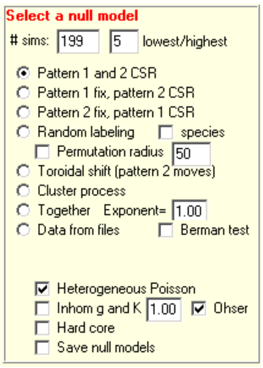


3.2.4 Heterogeneous Poisson with intensity from file

Example Book_Fig2_28h_file.res, intensity from file

This example repeats the previous analysis of Figure 2.28 using a heterogeneous Poisson process, but now uses an intensity function that was saved as a *. int file.

1. Execute *Programita*.
2. Highlight data file Book_Fig2_26.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Pattern 1 and 2 CSR**" in the window "Select a null model".
9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
10. Click checkbox "**Heterogeneous Poisson**"



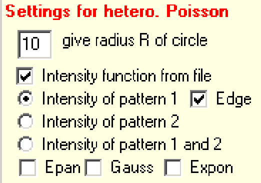
Select a null model

sims: 199 5 lowest/highest

● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test

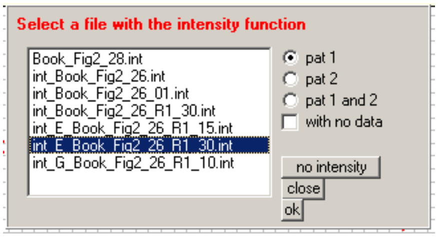
☑ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

11. Go to window "Settings for hetero. Poisson" on the left and click checkbox "**Intensity function from file**". Highlight file int_E_Book_Fig2_26_R1_30.int and press the small **ok**.



Settings for hetero. Poisson

10 give radius R of circle
☑ Intensity function from file
● Intensity of pattern 1 ☑ Edge
○ Intensity of pattern 2
○ Intensity of pattern 1 and 2
□ Epan □ Gauss □ Expon

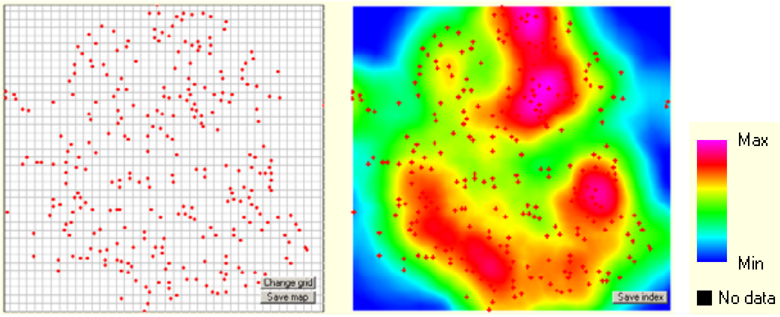


Select a file with the intensity function

Book_Fig2_28.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_Book_Fig2_26_R1_30.int
int_E_Book_Fig2_26_R1_15.int
int_E_Book_Fig2_26_R1_30.int
int_G_Book_Fig2_26_R1_10.int

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

and *Programita* shows you the data on the left and the intensity function together with the data on the right.



12. Click OK at the message box and then "**Calculate Index**" and *Programita* conducts the analysis.

3.2.5 Irregularly shaped observation window

Programita offers several options to analyze a univariate pattern within an observation window of irregular shape.

1. The points of the null model are only distributed inside the observation window, but otherwise no adjustments are done.
2. The observation window is explicitly reduced and the estimators of the second-order summary functions take the reduced observation window into account. The points of the null model are only distributed inside the observation window.
3. Inhomogeneous summary functions are used which are based on an intensity function $\lambda(\mathbf{x})$ which has a value of λ inside the observation window and zero outside (in the standard mode option 2 is implemented as option 3).

In the following example I show the first option which is based on the heterogeneous Poisson process and an intensity function which is zero outside the observation window and λ inside the observation window. In this case the CSR null model rejects points outside the observation window because they have a zero intensity and as a result the null model is CSR inside the observation window and no point of the null model will be located outside the observation window.

The **intensity file** must be an ASCII file with the *.int extension:

```
1  197  1  190  37430  1
  1  1  1  -9
  1  2  1  -9
  1  3  1  -9
  1  4  1  -9
  1  5  1  -9
  1  6  1   1
  1  7  1   1
  1  8  1  -9
  1  9  1  -9
  1  10 1  -9
….
```

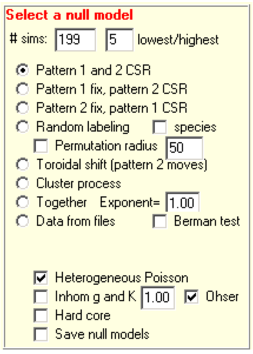
The file must describe a matrix and has therefore coordinates of a grid that run from 1 to 197 (x-coordinate) and 1 to 190 (y-coordinate). Thus, we have in total 197 × 190 = 37430 cells. Thus, the first line gives the first and last x-coordinate and the first and the last y-coordinate and the number of cells which follow. The last number is the cell size (i.e., 1 in the example).

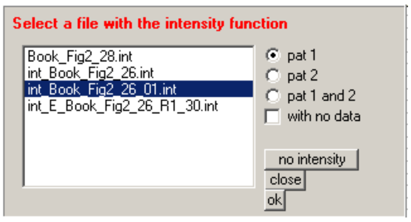
The following lines give the coordinates of all cells and its value. Each cell can have a value of -9 if it is outside the observation window or 1 if it is inside the observation window. (Note that the *.int files are always normalized between 0 and 1).

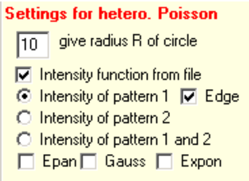
- columns 1 and 2: coordinates of the cells
- column 3: always 1
- column 4: value of normalized intensity function

Example Book_Fig2_28_opt1.res (manipulate null model, option 1)

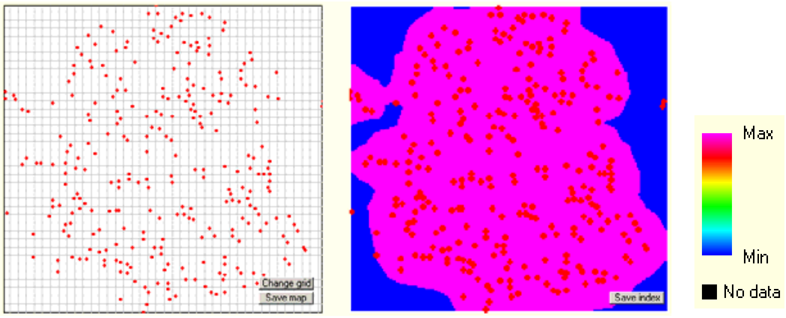
1. Execute *Programita*.
2. Highlight data file Book_Fig2_26.dat in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button “**Calculate Index**”
7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left.
8. Select “**Pattern 1 and 2 CSR**” in “Select a null model”.
9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
10. Click checkbox “**Heterogeneous Poisson**”
11. Go to window “Settings for hetero. Poisson” on the left and click checkbox “**Intensity function from file**”. Highlight file int_Book_Fig2_26.int and press the small **ok**.


Select a null model
# sims: 199 5 lowest/highest
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test
☑ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Obser
□ Hard core
□ Save null models


Select a file with the intensity function
Book_Fig2_28.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_E_Book_Fig2_26_R1_30.int
● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

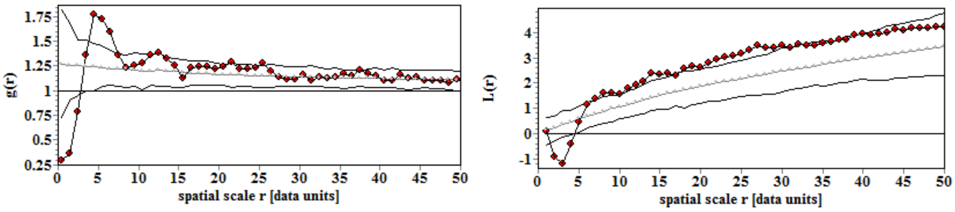

Settings for hetero. Poisson
10 give radius R of circle
☑ Intensity function from file
● Intensity of pattern 1 ☑ Edge
○ Intensity of pattern 2
○ Intensity of pattern 1 and 2
□ Epan □ Gauss □ Expon

and *Programita* shows you the data on the left and the intensity function together with the data on the right:



Click OK at the message box and then “**Calculate Index**” and *Programita* conducts the analysis. You can observe during the simulations that the null model does indeed not distribute points outside the observation window.

12. The result resembles that in Figure 2.28 e, f well:



Now I show the second option to consider observation windows of irregular shape which is based on an **explicit reduction of the observation window**. In this case the estimator of the second-order summary functions takes the reduced observation window into account and distributes the points of the null model only inside the observation window.

Two methods are available in the literature to consider observation windows of irregular shape.

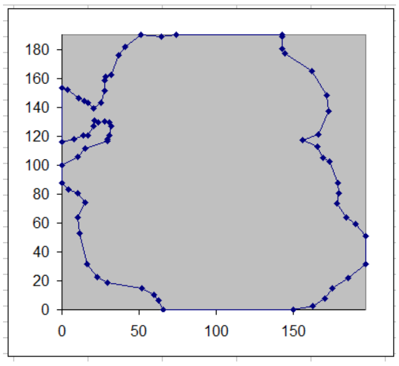
- First, Goreaud and Pélissier (1999) developed explicit equations for the Ripley edge correction for irregular observation windows where the geometrical shape is approximated by removing triangular surfaces from an initial rectangular shape. This method is not used in *Programita*.
- Second, Wiegand and Moloney (2004) approximated the geometrical shape of the observation window with an underlying grid and based the estimators of the second-order summary functions on estimates of the mean number of points in (potentially incomplete) rings (or circles) around the points of the focal individuals and the mean area of the rings (or circles) inside the observation window. For more details on this method see section 3.1.2.2 “*O*-Ring Statistic” and equation 3.36 in section 3.1.2.7 “Ripley’s *K*-Function”. This method is used in the standard grid based mode in *Programita*.
- Wiegand and Moloney (2014) extended the WM and Ohser estimators of the second-order summary functions to inhomogeneous estimators (see section 3.1.2.6 “Alternative Estimators of Inhomogeneous Pair-Correlation Functions” and equations 3.41 and 3.42 in section 3.1.2.7 “Ripley’s K-Function”). These estimators can be used to consider observation windows of irregular shape by using the intensity function introduced above which is zero outside the observation window and λ inside the observation window. This method is used for univariate analyses in the standard mode in *Programita*.

You need to tell *Programita* the shape of the observation window of irregular shape. This is done with a file that contains the coordinates of the border of the observation window which must result in a closed line.

The file with the border of the observation window must be an ASCII file with the *.irr extension. **Note that this is not an ArcGis shape file** but an ASCII file with the *.irr extension. In the example of Figure 2.28 it looks like this:

The first line gives the number of points and the following lines are the coordinates of the border. Note that the points must yield a closed curve and that the first and the last points must be the same.

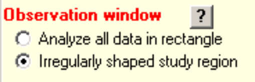
```
 70
 0.0    153.3
 3.4    152.2
10.8    146.1
14.2    144.3
17.1    143.0
20.6    139.0
25.6    143.0
27.7    151.2
28.3    160.7
32.0    162.3
….
```



Because *Programita* has only the coordinates of the curve, it needs some information what is inside and what is outside the observation window. To help *Programita* in this task, place a point of the pattern which is located well in the center of the observation window to the beginning of the *.dat data file. *Programita* uses this point as starting point to define the cells that belong to the observation window.

Book_Fig2_28_opt2.res (manipulate estimators, option 2)

1. Execute *Programita*.
2. Highlight data file Book_Fig2_26.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the radio button "**Irregularly shaped study region**" in the menu "Observation window" on the top left of the interface.

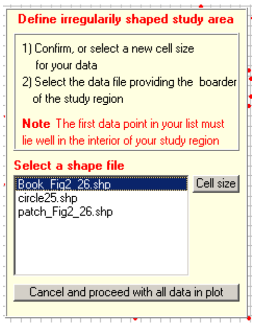


Observation window ?

○ Analyze all data in rectangle

● Irregularly shaped study region

8. Select file Book_Fig2_26.irr, click "**cell size**" and **ok** if the cell size appearing in the window "Select a new cell size" is ok and then the small **ok** button in the Select a shape file window. *Programita* now determines the area of the rectangle that belongs to the observation window. Basically, *Programita* generates an underlying grid with a spatial resolution of one bin (i.e., the cell size) and all cells outside are marked and excluded. *Programita* outputs the resulting intensity file as temporary file "int_temp.int".



Define irregularily shaped study area

1) Confirm, or select a new cell size for your data

2) Select the data file providing the boarder of the study region

Note The first data point in your list must lie well in the interior of your study region

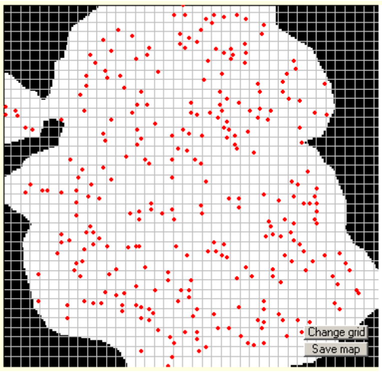
Select a shape file

Book_Fig2_26.shp
circle25.shp
patch_Fig2_26.shp

Cell size

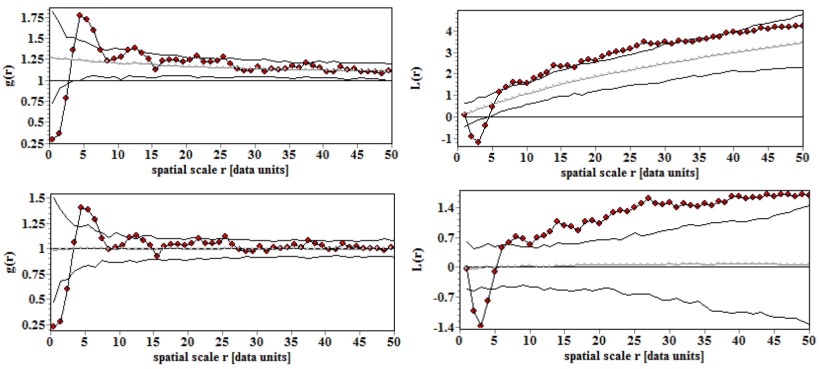
Cancel and proceed with all data in plot

9. Click "**Calculate Index**" and *Programita* shows a plot of the data within the reduced observation window. The excluded area is marked in black.



10. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
11. Select "**Pattern 1 and 2 CSR**" in the window "Select a null model".
12. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
13. Click the checkbox "**Calculate simulation envelopes** and *Programita* conducts the analysis. You can observe during the simulations that the null model does indeed not distribute points outside the observation window.

14. The results are different from that of the first option because *Programita* considers explicitly the shape of the observation window by estimation of the second-order summary functions. Top row: previous results (first option). Bottom row: results (second option):



Both, the pair correlation function and the *L*-function are now centered on the expectation of the CSR null model. This is because the estimators consider only the area of the observation window and the edge correction removed the bias seen in the first method that was caused by the heterogeneity of the pattern (i.e., the large patch in the center of the rectangle).

15. You can repeat the entire analysis also for the grid-based standard mode. The only difference is that you need to disable the option "**no grid**" in Which method will you use. As shortcut you can load the results file Book_Fig2_28_opt2.res, disable "**no grid**", and press "**Calculate Index**".

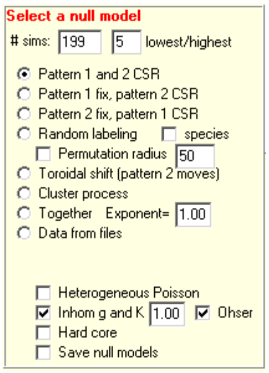
Try also the analogous analysis of Figure 3.48 based on Book_Fig3_48.dat, Book_Fig3_48.irr, and Book_Fig3_48c.res.

Finally, I show below how the third option based on inhomogeneous summary functions works in the standard mode. Note that the internal estimations of *Programita* are identical to that of the second option. To simplify the procedure for the user and to make it completely analogous to the grid-based mode, I programmed the second option in a way that *Programita* uses inhomogeneous summary functions.

In the second option you need to provide a polygon (i.e., the *.irr file) to define the observation window and *Programita* internally converts the polygon into an intensity function which is zero outside the observation window and λ inside the observation window (this intensity function is saved as temporary file "int_temp.int"). However, in the third option you need to provide the intensity function which is the same as used in the first option to condition the null model.

Example Book_Fig2_28_opt3.res (manipulate estimators, option 3)

1. Rename temporary file “int_temp.int” into “Book_Fig2_28.int”
2. Execute *Programita*.
3. Highlight data file Book_Fig2_26.dat you want to analyze in Input data and click the small “ok” button.
4. Select bin of 1m window Select a new cell size
5. Select a ring width of 3 in the menu “Which method will you use”
6. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
7. Press button “**Calculate Index**”
8. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
9. Select “**Pattern 1 and 2 CSR**” in the window “Select a null model”.
10. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
11. Enable checkbox “**Inhom g and k**”, highlight in the appearing window “Select a file with the intensity function” the intensity file **Book_Fig2_28.int**, and click the small **ok** button.

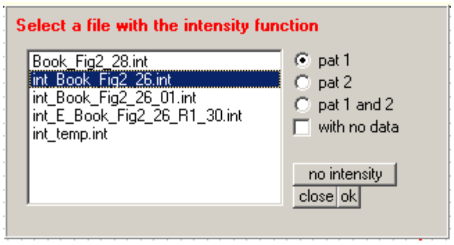


Select a null model

sims: 199 5 lowest/highest

● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

□ Heterogeneous Poisson
☑ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

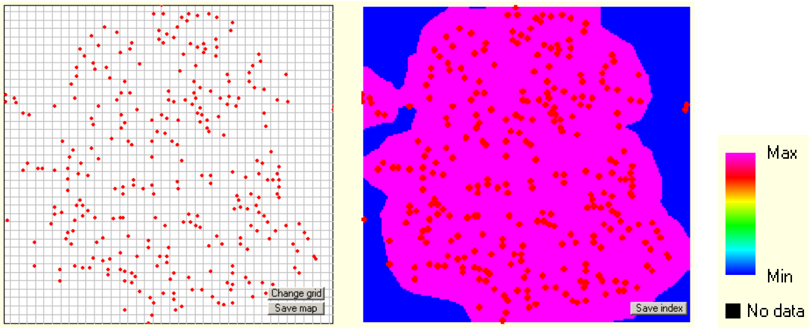


Select a file with the intensity function

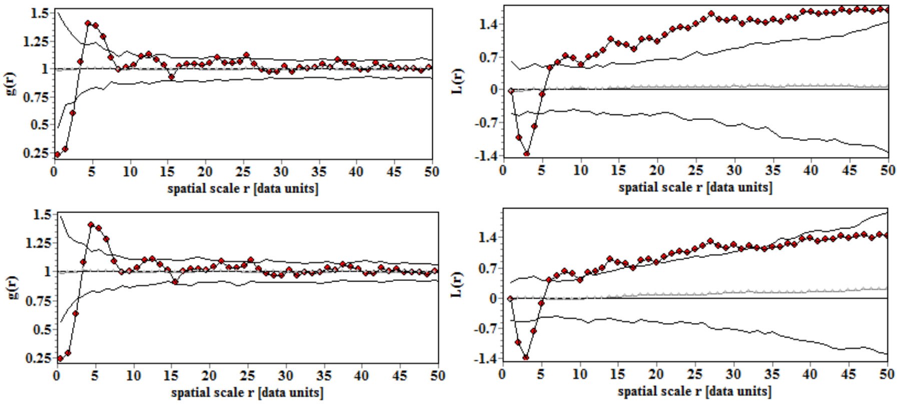
Book_Fig2_28.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_E_Book_Fig2_26_R1_30.int
int_temp.int

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

Because we analyze here univariate patterns the radiobox “pat 1” must be selected. This means that the intensity is assigned to pattern 1. *Programita* shows you the data on the left and the intensity function together with the data on the right:



14. Click OK at the message box and then “**Calculate Index**” and *Programita* conducts the analysis. You can observe during the simulations that the null model does indeed not distribute points outside the observation window.
15. The results are virtually identical to that of option 2. Top row: previous results (second option). Bottom row: results (third option):



Both, the pair correlation function and the *L*-function are now centered on the expectation of the CSR null model. This is because the estimators consider only the area of the observation window and the edge correction removed the bias seen in the first method that was caused by the heterogeneity of the pattern (i.e., the large patch in the center of the rectangle).

3.2.6 Null model from file

In some cases you may not generate the null model patterns internally with *Programita*, but use patterns generated from other sources for this purpose. One important example for this case is pattern reconstruction (section 4.1.3 "Null Model of Pattern Reconstruction" in Wiegand and Moloney 2014). In this case you can generate from a given pattern statistical replicates that are optimized to closely match several summary functions of the observed pattern. Of course, the reconstructed patterns will not be an identical copy of the observed pattern, but show the same statistical features as the observed pattern where the typical structures will appear at somewhat displaced locations.

Be sure to use the same estimator for the second-order summary functions in pattern reconstruction and in *Programita*.

The **first method** of the pattern reconstruction software presented in Wiegand et al. (2013) uses the Ohser edge correction (see equation 3.9 in Wiegand and Moloney 2014) and two times the natural estimator of the intensity $\lambda_n = n/A$ (i.e., the non-adapted intensity estimators):

$$g(r) = \frac{1}{\lambda_n}\frac{1}{\lambda_n}\frac{1}{2\pi\, r}\sum_{i=1}^{n}\sum_{j=1}^{n,\neq} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)[\frac{1}{\bar{\gamma}_W(r)}]$$

This method is used because it corresponds to pattern reconstruction without edge correction for $g(r)$ because the term $A/\bar{\gamma}_W(r)$ does not depend on the points pair i, j but only on distance r and can therefore be factored out in the estimation of the partial energy (equation 3.317 in Wiegand and Moloney 2014).

Method 2 in the pattern reconstruction software corresponds to the WM estimator in *Programita*:

$$g(r) = \frac{1}{\lambda_n}\frac{1}{2\pi\, r}\sum_{i=1}^{n}\sum_{j=1}^{n,\neq} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)[\frac{2\pi\, r}{\sum_{i=1}^{n} \nu_{d-1}(W \cap \partial b(\mathbf{x}_i, r))}]$$

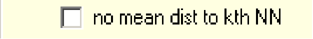
which results from replacing one of the natural estimators λ_n of the intensity in method 1 by the adapted intensity estimate

$$\hat{\lambda}_S(r) = \frac{\sum_{i}^{n} \nu_{d-1}(W \cap \partial b(\mathbf{x}_i, r))}{2\pi\, r\, \bar{\gamma}_W(r)}$$

and the **method 3** in the pattern reconstruction software corresponds to the Ohser estimator in *Programita* with adapted intensity estimate where both natural estimators λ_n of the intensity in method 1 are replaced by the adapted intensity estimate $\lambda_S(r)$

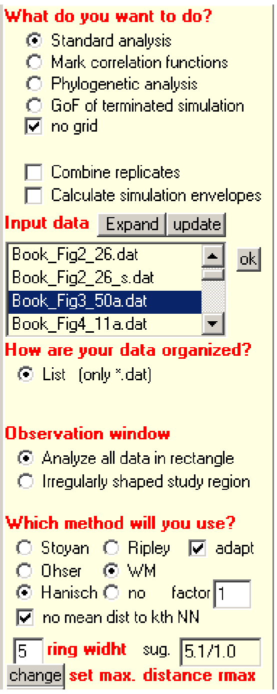
Example Book_Fig3_51.res (null model from file)

1. Execute *Programita*.
2. Highlight data file Book_Fig3_50a.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 5 in the menu "Which method will you use"
5. Click button "change" below to set maximal distance *r* to be analyzed. Insert 100 in small box that opens and then the small **ok** button.
6. To view a large range of neighborhood ranks *k* in the mean distance to the kth neighbor summary function *nn*(*k*), disable the option "**no mean distance to kth NN**"

   □ no mean dist to kth NN

   .
7. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
8. Press button "**Calculate Index**"
9. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
10. Select "**Data from file**" in the window "Select a null model".



What do you want to do?

● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

□ Combine replicates
□ Calculate simulation envelopes

Input data Expand update

Book_Fig2_26.dat
Book_Fig2_26_s.dat
Book_Fig3_50a.dat
Book_Fig4_11a.dat
ok

How are your data organized?

● List (only *.dat)

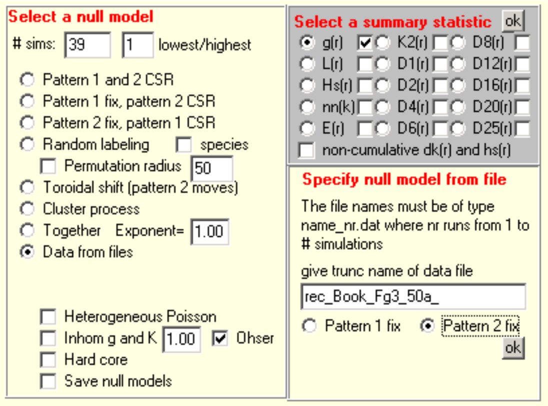
Observation window

● Analyze all data in rectangle
○ Irregularly shaped study region

Which method will you use?

○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
● Hanisch ○ no factor 1
☑ no mean dist to kth NN

5 ring widht sug. 5.1/1.0
change set max. distance rmax



Select a null model

sims: 39 1 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
● Data from files

□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

Select a summary statistic ok

● g(r) ☑ ○ K2(r) □ ○ D8(r) □
○ L(r) □ ○ D1(r) □ ○ D12(r) □
○ Hs(r) □ ○ D2(r) □ ○ D16(r) □
○ nn(k) □ ○ D4(r) □ ○ D20(r) □
○ E(r) □ ○ D6(r) □ ○ D25(r) □
□ non-cumulative dk(r) and hs(r)

Specify null model from file

The file names must be of type name_nr.dat where nr runs from 1 to # simulations

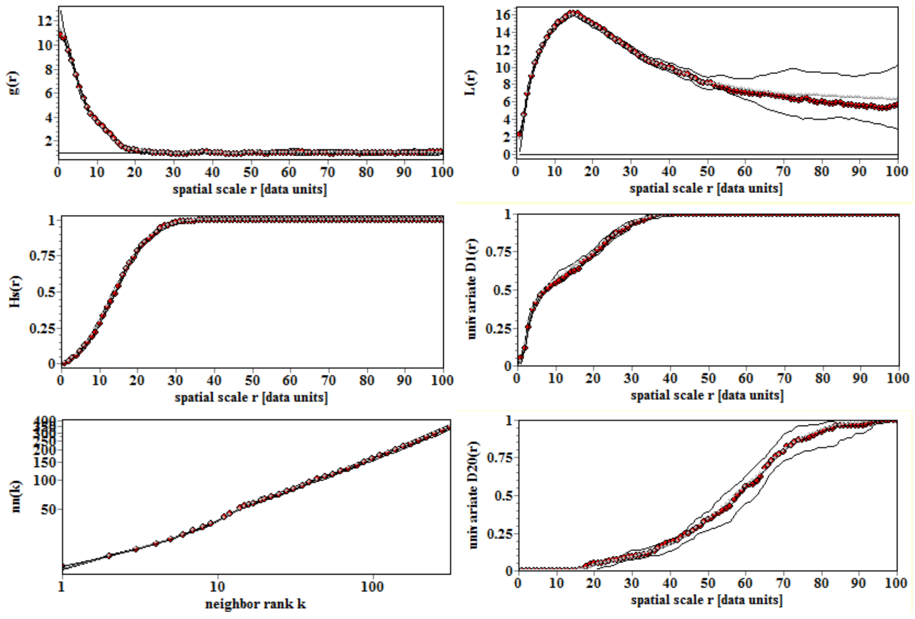
give trunc name of data file

rec_Book_Fg3_50a_

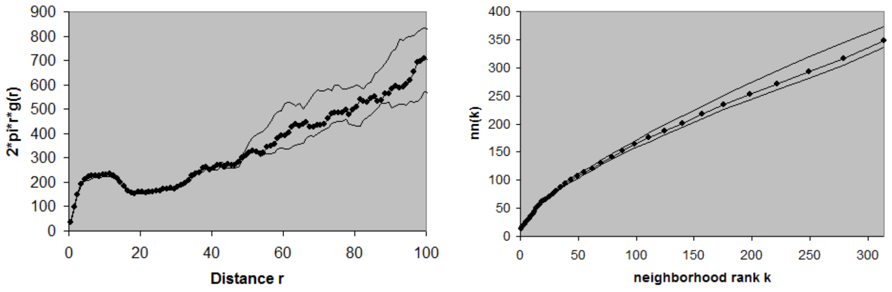
○ Pattern 1 fix ● Pattern 2 fix
ok

11. Insert the trunk name of the null model files (rec_Book_Fig3_50a_) in the window "Specify null model from file" that opens. This is because your data file had the name "Book_Fig3_50a.dat" and because the pattern reconstruction software names the reconstructions rec_name_n.dat where the "rec_" indicates that this is a reconstructed data file and the n is the number of the reconstructions. Thus you have null model files rec_Book_Fig3_50a_1.dat, rec_Book_Fig3_50a_2.dat, …

12. Click also the radio button “**Pattern 2 fix**”. This means that the null model files are used for pattern 1. In bivariate analysis you will typically leave pattern 1 unchanged but replace pattern 2 in the null model by pattern reconstruction files and therefore select “Pattern 1 fix”. To finish click the small **ok** button in the window “Specify null model files from file”.Specify the number of simulations of the null model (19 in the example) and the rule for the estimation of simulation envelopes (here the 1th lowest and highest values of the summary function of the 19 simulated null model data sets).
13. If all settings are specified, press the button “**Calculate Index**” and *Programita* conducts the simulations of the null model.
14. In window “Select a summary function” you can view the results of the analysis for the different summary functions and compare with Figure 3.51 in Wiegand and Moloney (2014):



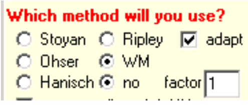
If you take the results for the pair correlation function from the *.res file and estimate (2π *r*)*g*(*r*) from of *g*(*r*) and plot (2π *r*)*g*(*r*) over r and *nn*(*k*) non-logarithmically you find:

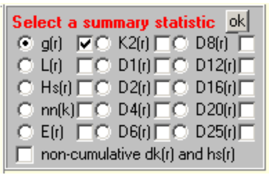


3.2.7 Combine replicates for standard analysis

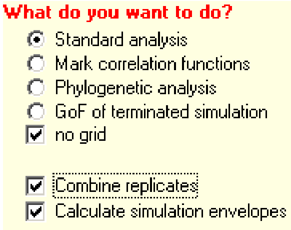
In some cases you may have maps of several replicate plots of a larger point pattern under identical conditions. In this case the resulting test statistics of the individual replicate plots can be combined into average test statistics (Diggle 2003: page 123; Illian et al. 2008: page 263; Wiegand and Moloney 2014: section 3.2). This is of particular interest if the number of points in each replicate plot is relatively low. In this case the simulation envelopes of individual analyses would become wide, but combining the data of several replicate plots into average test statistics increases the sample size and thus narrows the simulation envelopes. Section 3.2.1 of Wiegand and Moloney (2014) provides details on the aggregation formulas for different summary functions and section 3.2.2 several examples.

The default estimators of the *Programita* standard mode (and the grid-based mode) use the WM estimators for the pair correlation and the *K*-function based on the quantities $\lambda g(r)$ and $\lambda K(r)$. The corresponding aggregation formulas for the WM estimator are provided in equations 3.114 and 3.117 in Wiegand and Moloney (2014). If you select other summary functions than $g(r)$ or $L(r)$, the aggregation formula used by *Programita* is the abundance weighted mean of the summary functions of the individual replicates where the weight is the abundance of the focal pattern 1. Note that it makes little sense to apply the aggregation formula to the K2 function, here you should first estimate the pair correlation function and then estimate the derivative.


Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1

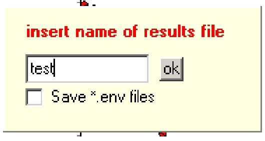

Select a summary statistic ok
● g(r) ☑ ○ K2(r) □ ○ D8(r) □
○ L(r) □ ○ D1(r) □ ○ D12(r) □
○ Hs(r) □ ○ D2(r) □ ○ D16(r) □
○ nn(k) □ ○ D4(r) □ ○ D20(r) □
○ E(r) □ ○ D6(r) □ ○ D25(r) □
□ non-cumulative dk(r) and hs(r)

Before running an individual standard analysis enable the checkbox "**Combine replicates**", then run all analyses with replicate plots of the same treatment with the same settings (this is important!) and the same summary function(!), i.e., do not change the maximal scale analyzed, or the grid size, or the summary function. This can be done most conveniently with the "**Series of analyses**" option described above.


What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

☑ Combine replicates
☑ Calculate simulation envelopes

When the option "**Combine replicates**" is enabled, *Programita* creates specific results files that contain all information necessary for combining the replicates. If your data file was named "name.dat", the corresponding results file will be called "WM_name.rep" if you used the pair correlation function (or any other summary function in the standard mode other than the *L*-function) and "R_name.rep" if you select the *L*-function. However, better save the data using the Save results option: Save results for example with the name "test"


insert name of results file
test ok
□ Save *.env files

In this case Programita saves the two files test.res and test.rep that contain all information needed to combine the results of several replicate analyses.

This is a typical WM_name.rep output file:

```
      3     199   626      0 250000  W-M     1  gridless  g(r)
 0   0    1966.6370    100.0000      0.0000  12732.40    0.00
 0   1    5848.9918    290.0000      0.0000  12415.13    0.00
 0   2    9639.6005    354.0000      0.0000  9195.57    0.00
 0   3   13395.3274    374.0000      0.0000  6991.21    0.00
 1   0    1966.6370      2.0000      0.0000  254.65    0.00
 1   1    5878.7354     10.0000      0.0000  425.94    0.00
 1   2    9770.7399     14.0000      0.0000  358.79    0.00
 1   3   13644.6155     30.0000      0.0000  550.55    0.00
 2   0    1966.6370      4.0000      0.0000  509.30    0.00
 2   1    5885.8916     16.0000      0.0000  680.68    0.00
 2   2    9774.6603     34.0000      0.0000  870.99    0.00
 2   3   13663.3306     26.0000      0.0000  476.49    0.00
 C1  C2   C3            C4            C5      C6          C7
```

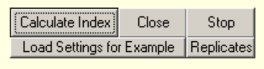
The header contains basic information on the number of distance bins used (3), the number of simulations of the null model (199), the number n_1 of points of pattern 1 (626), the number n_2 of points of pattern 2 (0), the area of the observation window in units of the bin (250000), and if the L-function (R) or any other summary function was selected (W-M). Additionally, in the standard mode the header contains information on the test function selected [1 and g(r)], and (gridless). The following columns contain information of the estimators:

- C1: number of simulation of the null model where 0 are the observed data and 1, 2, are the simulations of the null model.
- C2: the distance bin
- C3: the denominator of equation (3.106) in Wiegand and Moloney (2014). If an estimator other than WM is selected (i.e., Ohser, Stoyan or Ripley), the denominator yields n_1 times the area of a ring with radius r and width 1 [$n_1 2\pi r$] for the pair correlation function, and n_1 times the area of a circle with radius r [$\pi\ r^2$] if the L-function is selected. (Note that the bins are 0.5, 1.5, 2.5, … for the pair correlation function and 0, 1, 2, 3,… for the L-function)

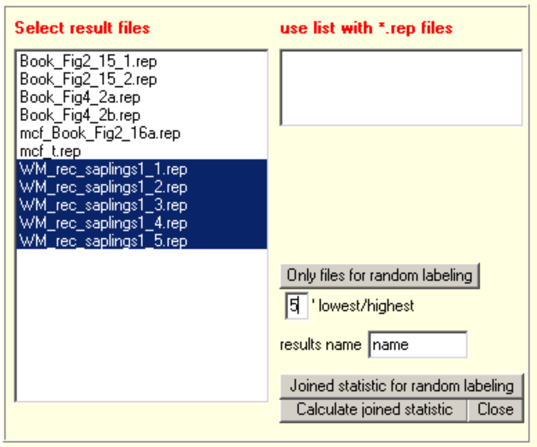
- C4: the numerator of equation (3.106) in Wiegand and Moloney (2014). If an estimator other than WM is selected, the numerator yields $n_1 2\pi r\lambda g(r)$ for the pair correlation function, and $n_1 \pi r^2 \lambda K(r)$ if the L-function is selected. Application of the aggregation formulas 3.114 and 3.117 in Wiegand and Moloney (2014) then yield the n_1 weighted mean of the summary functions of the individual replicates.
- C5: same as C4, but for the bivariate summary function.
- C6: n_1 times the summary function selected, for example $n_1\ D^k{}_{11}(r)$ if you selected the distribution function to the kth neighbour.
- C7: same as C6, but for the bivariate summary function.

The columns C1-C5 are the same for all summary functions selected, columns C6 and C7 contain the information on the summary functions other than $g(r)$ and $L(r)$.

Once you completed all analyses close *Programita*, open it again and click the button "**Replicates**" below the "Stop" button:

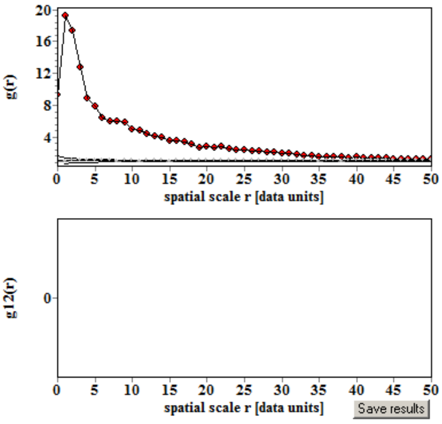


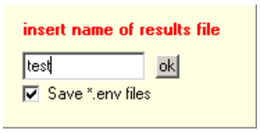
A window opens where you can select the files you like to combine. Highlight the files you want to combine and click "**Calculate joined statistic**", and the result of the combined analysis appears.



5 ' lowest/highest

results name: name





test

☑ Save *.env files

You can save the results using the "**Save results button**". Insert a name. If "name" stands for the selected name, the results file "name.res" gives you the mean weighted summary function, the file "name.rep" the file that allows you to view the results with "**Combine replicates**", and *.env files with the summary functions for the observed data and the null model simulations for use in the GoF test.

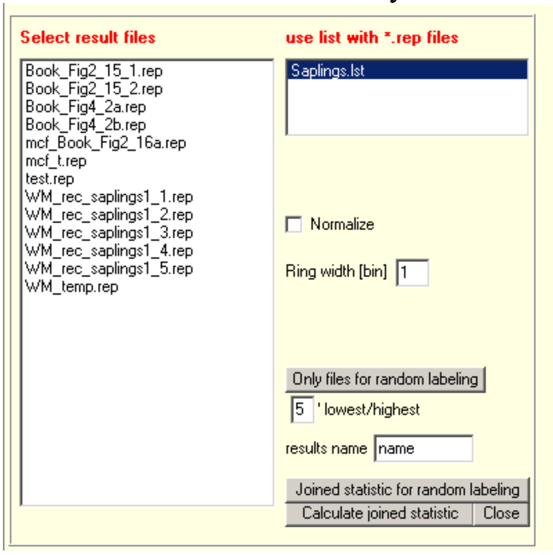
Note that these *.rep and *.env files contain at the end a list of the files you combined. This result file joined the results from point-pattern analysis of several single analyses in different replicate plots, namely:

```
This result file joined the results from point-pattern analysis of several single analyses in
different replicate plots, namely:
WM_rec_saplings1_1.rep
WM_rec_saplings1_2.rep
WM_rec_saplings1_3.rep
WM_rec_saplings1_4.rep
WM_rec_saplings1_5.rep
```

Instead of highlighting individual *.rep files in the listbox on the left (Select result files), you can also select a file that contains a list with the names of the results files you want to combine (Use list with *.rep files). This file must be an ASCII file with the *.lst extension. The file corresponding to the example above is named Saplings.lst and is:

```
5
WM_rec_saplings3_1.rep
WM_rec_saplings3_2.rep
WM_rec_saplings3_3.rep
WM_rec_saplings3_4.rep
WM_rec_saplings3_5.rep
```

where first line is number of files and following lines are the names of the *.rep files you want to combine. This allows you to combine files with certain criteria in a quick way.



Select result files

Book_Fig2_15_1.rep
Book_Fig2_15_2.rep
Book_Fig4_2a.rep
Book_Fig4_2b.rep
mcf_Book_Fig2_16a.rep
mcf_t.rep
test.rep
WM_rec_saplings1_1.rep
WM_rec_saplings1_2.rep
WM_rec_saplings1_3.rep
WM_rec_saplings1_4.rep
WM_rec_saplings1_5.rep
WM_temp.rep

use list with *.rep files

Saplings.lst

□ Normalize

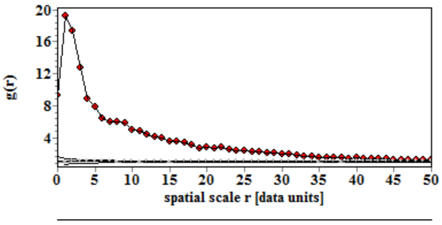
Ring width [bin] 1

Only files for random labeling

5 ' lowest/highest

results name name

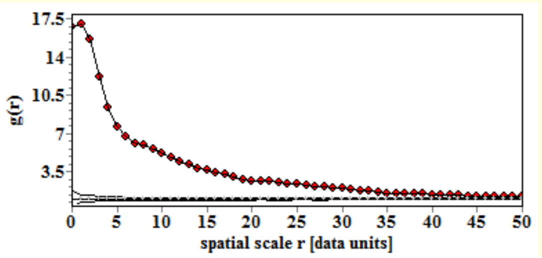
Joined statistic for random labeling
Calculate joined statistic
Close



Change ring width for pair correlation function

If you use the pair correlation function as summary function, this feature of *Programita* allows you also to change a posteriori the ring width. However, **to take advantage of this feature, the original analyses must be done with ring width of 1**: 1 **ring widht**. Then you can change the ring width in the Combine replicated window, as shown above, using

Ring width [bin] 3



To select a posteriori a wider ring width, go again to Replicates (without closing *Programita*) and select again the same files. If the ring width of the original files was 1, the box Ring width [bin] 3 appears where you can change the ring width. After changing the ring width go again to Calculate joined statistic to get the corresponding result seen on the left.

3.3 Homogeneous cluster processes

3.3.1 Overview on Thomas cluster processes

Thomas cluster processes are point process models that describe clustering in a simple way. Details can be found in section 4.1.4 “Poisson Cluster Point Processes” in Wiegand and Moloney (2014). *Programita* allows you to fit several Thomas cluster processes to univariate data:

1. a simple Thomas process with one critical scale of clustering
2. a simple bivariate parent-offspring Thomas process where the cluster centers are known
3. a bivariate parent-offspring Thomas process where the known cluster centers are themselves clustered
4. a generalized Thomas process with two nested scales of clustering where small clusters are located inside large clusters.

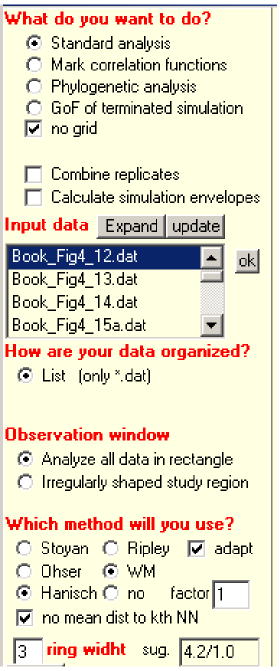
Additionally, the cluster processes can be independently superimposed with a CSR pattern and you can change the distribution of points over the clusters from a Poisson distribution (random assignment to a cluster) to a negative Binominal distribution (clustered assignment).

The cluster processes are parametric point processes and must be fitted to the data. *Programita* allows you to do so in a straight forward way. The procedure to fit a cluster process to the data is described in sections 2.5.2.1 “Minimum Contrast Methods” and 4.1.4.3 “Fitting a Thomas Process to the Data” in Wiegand and Moloney (2014). *Programita* allows you to conduct the fit of the simple (1) and the double-cluster Thomas process (4) in an automated way, thereby facilitating the automated analysis of several patterns. Additionally, *Programita* uses a specific technique to avoid that departures from the point process model at small scales “contaminate” the fit with the cumulative *K*-functions at larger scales. This is an important advance over current techniques and avoids a bias in the fitted parameters. This is exemplified in the first example.

3.3.2 Thomas cluster processes with one scale of clustering

This example illustrates the procedure in *Programita* that allows manually fitting of a cluster process to a point pattern. It also shows how to deal with patterns that show an additional pattern at small scales not accommodated by the cluster process.

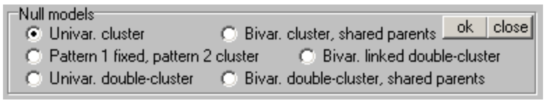
1. Execute *Programita*.
2. Highlight data file Book_Fig4_12.dat you want to analyze in Input data and click the small “ok” button. This data file was generated with a nested double cluster Thomas process with parameters of large scale clustering being σ_1 = 8.69 and $A\rho_1$ = 68.7 clusters. The additional small-scale clustering had parameters σ_1 = 2.64 and $A\rho_1$ = 147 clusters.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks of $D^k(r)$.
6. Press button “**Calculate Index**”
7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left.


What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid
☐ Combine replicates
☐ Calculate simulation envelopes
Input data

Book_Fig4_12.dat
Book_Fig4_13.dat
Book_Fig4_14.dat
Book_Fig4_15a.dat

How are your data organized?
● List (only *.dat)
Observation window
● Analyze all data in rectangle
○ Irregularly shaped study region
Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
● Hanisch ○ no factor 1
☑ no mean dist to kth NN
3 ring widht sug. 4.2/1.0

8. Select “**Cluster process**” in the window “Select a null model”.
9. A window “Fit of cluster process to data” opens. Select in the section “Null models” at the bottom “**Univar. cluster**”. This is the simplest Thomas process with one critical scale of clustering. Continue with the small **ok** button



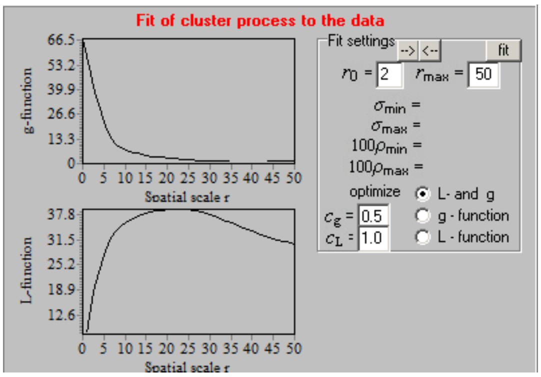
Null models

● Univar. cluster ○ Bivar. cluster, shared parents

○ Pattern 1 fixed, pattern 2 cluster ○ Bivar. linked double-cluster

○ Univar. double-cluster ○ Bivar. double-cluster, shared parents

10. Now the interface for fitting appears:



Fit of cluster process to the data

Fit settings

r_0 = 2 r_{max} = 50

σ_{min} =

σ_{max} =

$100\rho_{min}$ =

$100\rho_{max}$ =

optimize ● L- and g ○ g - function ○ L - function

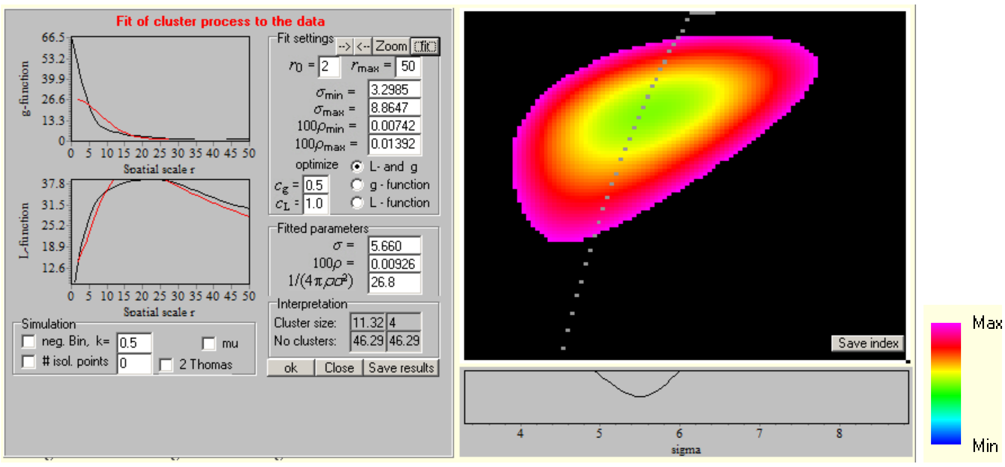
c_g = 0.5

c_L = 1.0

Select the radio button “**L-function**” to only use the *L*-function for fitting. The default settings now fits the L-function over distance interval 2 to 50m ( ).

r_0 = 2 r_{max} = 50

11. Click the button “**fit**” and *Programita* fits the two parameters ρ and σ of the Thomas process to the pattern. Note that ρA yields the number of clusters and 2σ the approximate radius of the “typical cluster”. To iteratively encircle the parameter space around the minimum in the σ-ρ parameter space click “**Zoom**” and “**Fit**”:



Fit of cluster process to the data

Fit settings

r_0 = 2 r_{max} = 50

σ_{min} = 3.2985

σ_{max} = 8.8647

$100\rho_{min}$ = 0.00742

$100\rho_{max}$ = 0.01392

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 5.660

100ρ = 0.00926

$1/(4\pi\rho\sigma^2)$ = 26.8

Interpretation

Cluster size: 11.32 4

No clusters: 46.29 46.29

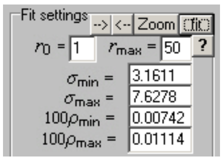
Simulation

□ neg. Bin, k= 0.5 □ mu

□ # isol. points 0 □ 2 Thomas

The graph on the right shows the deviation between observed summary function (here only the *L*-function) and that predicted by Thomas process over the σ-ρ parameter space indicated by σ_{min}, σ_{max}, $100\rho_{min}$, and $100\rho_{min}$. There is a clear minimum at σ = 5.7 and 46.3 clusters. However, the lower left graph show that the fit of the L-function is not satisfying. This corresponds to Figure 4.12a.

12. Instead of using the **Zoom** option to iteratively encircle the minimum in parameter space you can also manually change the σ-ρ parameter space by selecting appropriate values for σ_{min}, σ_{max}, $100\rho_{min}$, and $100\rho_{min}$:



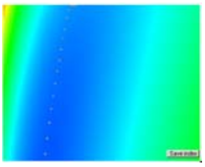
r_0 = 1 r_{max} = 50

σ_{min} = 3.1611

σ_{max} = 7.6278

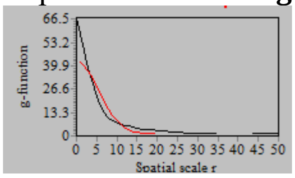
$100\rho_{min}$ = 0.00742

$100\rho_{max}$ = 0.01114

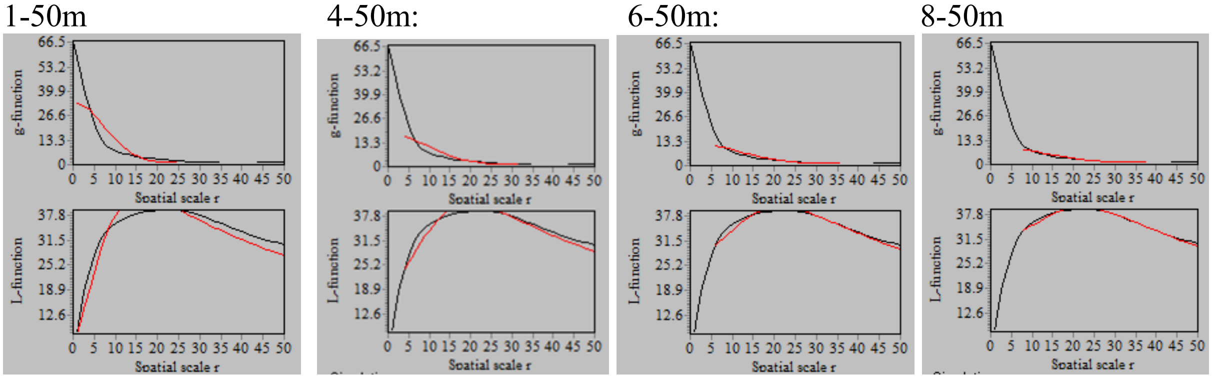


This may be required if the (initially) selected parameter space does not contain the minimum and the fit is poor. In this case a message "increase the maximal value of sigma", "increase the maximal value of roh", "decrease the minimal value of sigma" or "decrease the minimal value of roh" may appear. In this case the observed minimum is located at the edge of the selected parameter space (see right figure above).

13. To use only the pair correlation function for the fit over the 1 - 50m interval repeat steps 1-12 but click "**g - function**". Again, as in Figure 4.12b, the fit is not satisfying:

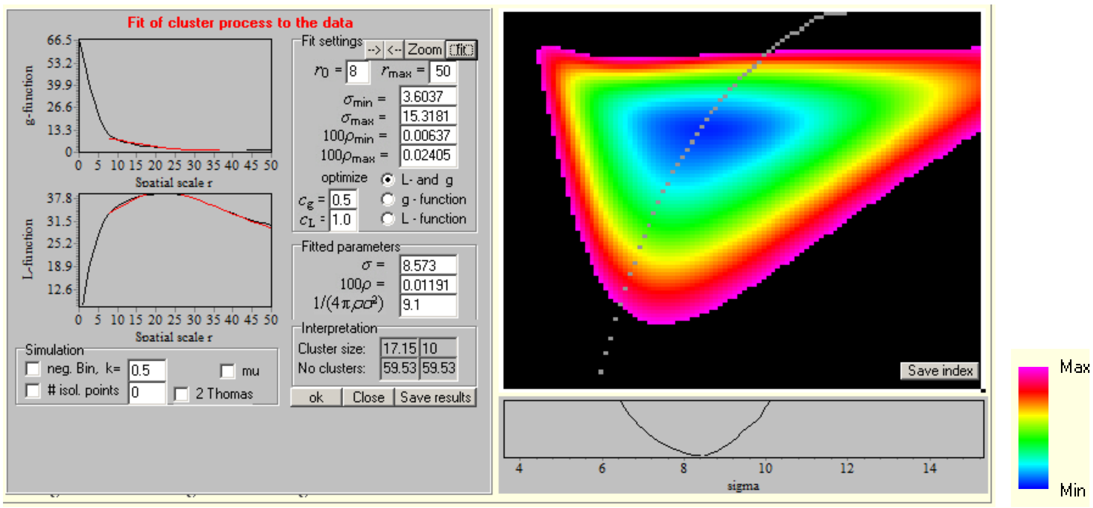


14. The important issue is that you recognize from the shape of the pair correlation function that the pattern contains probably a second critical scale of small-scale clustering. By manipulating the distance r_0 (i.e., the lower limit of the interval of the fit) you can determine the scale of small-scale clustering. For this use both, the pair correlation function and the *L*-function for fitting, i.e., select "**L- and g - function**":


1-50m 4-50m: 6-50m: 8-50m

15. The interval of 8-50m provides the best approximation. Thus, the contribution of the additional small-scale clustering to the pair correlation function disappears after 8m. Note that the transformation (equation 4.12) "attaches" the left value of the *L*-function [i.e., $L(r_0)$] to the observed value, thereby removing the "memory" of the *L*-function

Fitting the cluster process only for distances > 8m allows you to determine the parameters of the large-scale clustering. We obtain the parameter estimates σ_1 = 8.6 and 59.6 clusters which are very close to the parameters σ_1 = 8.69 and 68.7 clusters that were used to generate the pattern. The parameter space also indicates a deep and clear minimum (i.e., with dark blue).



Fit of cluster process to the data

Fit settings

r_0 = 8 r_{max} = 50

σ_{min} = 3.6037

σ_{max} = 15.3181

$100\rho_{min}$ = 0.00637

$100\rho_{max}$ = 0.02405

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 8.573

100ρ = 0.01191

$1/(4\pi\rho\sigma^2)$ 9.1

Interpretation

Cluster size: 17.15 10

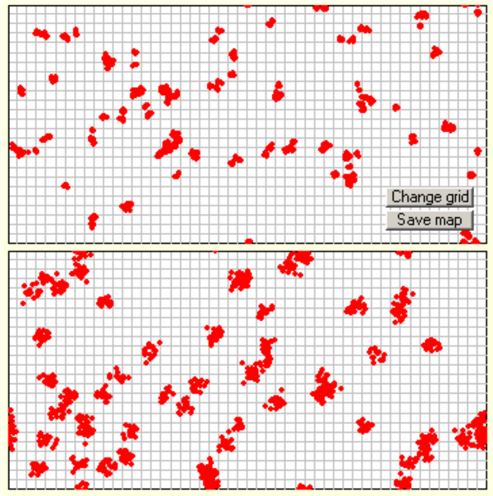
No clusters: 59.53 59.53

Simulation

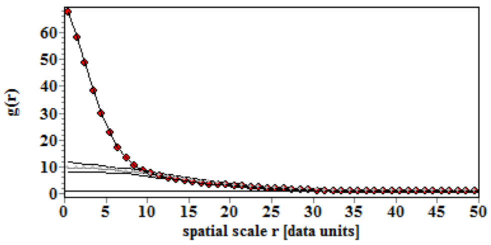
□ neg. Bin, k= 0.5 □ mu

□ # isol. points 0 □ 2 Thomas

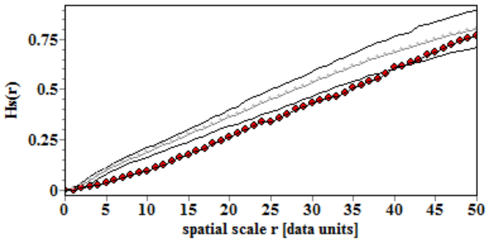
17. If you are satisfied with the fit, click the **ok** button ok Close Save results and then "**Calculate Index**". *Programita* now conducts the simulations of the fitted Thomas process:

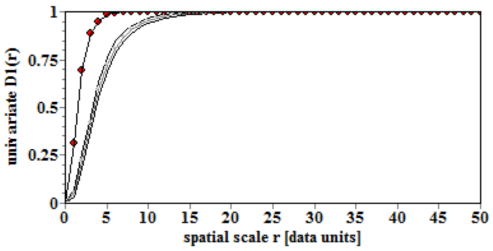


Here the top graph shows the observed pattern and the bottom graph the simulated patterns. As expected, the fitted point process fits the pair correlation function for distances *r* > 8m very well, but an additional clustering is visible at smaller distances:



The spherical contact distribution is overestimated which means that the gaps in the simulated pattern are too small and the distribution function of the distances to the nearest neighbor is underestimated which means that the nearest neighbor in the simulated patterns are in general too far away.





Clearly, this is because the additional small-scale clustering is missing where 147 small clusters are nested inside the 68.7 large clusters. Thus each large cluster contains on average 2.2 clusters with radius 2σ = 5.3m. For this reason we notice that the observed clusters look somewhat smaller than the simulated ones.

In step 16 you can also save the results of the parameter fitting. In this case a file Book_Fig4_12.fit is generated.

- the first part of the file shows the settings of the *.res file.
- the second part of the file show details on the fit such as the interval in parameter space, the best fitting parameters and their interpretation.
- the third part of the file contains the history of the fitting settings
- the fourth part of the file contains the observed and fitted *g*- and L-*f*unction.
- the final part of the file contains the final parameter space and the associated errors.

```
This file contains settings and results of fitting your data to a cluster1 Thomas cluster process ---------
---------------------------------------------------------------

Your settings for the point-pattern analysis were:

Pointpattern analysis of file T:\towi\thorsten\text\ManualProgramita2013\Programita\Book_Fig4_12.dat
Method Wiegand-Moloney (ring) with 199 replicates for simulation envelopes, ring width =   3   5 th lowest and highest values of   199 simulations
Test Model= cluster1   8.6270 0.00011900
the null model assumed homogeneous pattern(s)
Analysis modus= points        gridless    WM    NN Hanisch
several points per cell allowed
All cells within the rectangle were considered for calculating the indices
number points of pattern 1 = 1160
number points of pattern 2 =    0
the rectangular area contains 1000*500 = 500000   cells  (= dim1*dim2)
 x-grid-size= 1000 y-grid-size=  500 cell-size =  1.0000 units. rmax=     50,  max distance for NN functions:    279 mean dist to kth neighbor only
partly determined
---------------------------------------------------------------

 Your settings for the fit of the L-and g-function with a Thomas cluster process were:

 Interval (r0, rmax)   = (8,50)
 Interval for sigma    = (3.5844,13.9855)
 Interval for 100*roh = (0.00635,0.02299)
 The power transformations:
   c = 1.0  for L
   c = 0.5  for g
 The fitted parameters are:
   sigma    = 8.627
   100*roh = 0.01190 which corresponds to 59.48 parents in the study region and to 59.48 parents in the rectangle
 You optimezed the L- and g-function simultaneously
 Only fits with an error <0.02500 are accepted for the estimation of the confidence interval
 The error for the best fit was: 0.00068
 The confidence interval for sigma was:     ( 4.4249,13.9855). This is the interval where the error is <0.02500
 The confidence interval for 100*roh was:   ( 0.0080, 0.0210). This is the interval where the error is <0.02500
```

```
----------------------------------------------------------------
 Here is the setting history of your fit
step sigmin   sigmax   100romin   100romax     R0   rmax    power_g    power_L     sigma     100roh      g          error   L_g_Lg
12      3.58    13.99   0.006350   0.022990     8     50     0.5000     1.0000      8.63   0.011897       8.99     0.00068   3
11      3.49    11.65   0.006390   0.019160     8     50     0.5000     1.0000      8.60   0.011937       9.01     0.00069   3
10      3.49    11.65   0.006390   0.019160     6     50     0.5000     1.0000      7.94   0.011034      11.43     0.00208   3
 9      3.33     9.71   0.006340   0.015970     6     50     0.5000     1.0000      7.91   0.011009      11.56     0.00208   3
 8      3.33     9.71   0.006340   0.015970     4     50     0.5000     1.0000      6.81   0.010036      17.09     0.00613   3
…
```

```
---------------------------------------------------------------

Here are the data and the fits:
  r  Ldata   Lfit   gdata   gfit
  1   7.7842  33.1815 58.2340   9.9568
  2  14.4385  32.6138 48.5176   9.8670
  3  19.9061  32.3140 38.3178   9.7193
  4  24.3846  32.2546 29.6160   9.5167
  5  27.8025  32.4019 22.5303   9.2631
  6  30.4101  32.7187 16.9736   8.9633
  7  32.3811  33.1676 13.1916   8.6231
  8  33.7127  33.7127 10.4626   8.2486
  9  34.8369  34.3211  8.8404   7.8463
 10  35.6848  34.9637  7.4954   7.4230
 11  36.3997  35.6152  6.6477   6.9856
…
```

```
---------------------------------------------------------------
  Here are the best parameters together with the error: (Note that all errors > 0.02500 are set to the value 0.02500)
 sigma  100*roh     error
3.5844  0.00635  0.025000
3.5844  0.00652  0.025000
3.5844  0.00669  0.025000
3.5844  0.00685  0.025000
3.5844  0.00702  0.025000
```

3.3.3 Generalized simple Thomas processes

This example illustrates the generalization of the simple Thomas process with one scale of clustering that allows for clumping of the number of points over the clusters (see section 4.1.4.2. “Thomas process” in Wiegand and Moloney 2014).

Remember that the simple Thomas process assigned the points randomly to the clusters, thus yielding a Poisson distribution for the distribution p_S of the number of points S per cluster. However, if the distribution p_S follows a more general negative Binominal distribution with clumping parameter k, we can generate more realistic patterns where some clusters have more points than expected by random allocation of the points over the clusters, and others have less than expected. For $k \rightarrow \infty$ the negative Binominal distribution collapses to the Poisson distribution.

The nice feature of the Thomas process is that a negative Binominal distribution does not change the functional form of the analytical solution of the pair correlation function (and the K-function) of the Thomas process. We obtain:

$$g(r,\rho,\sigma) = 1 + \frac{f_k}{\rho}\frac{\exp(-r^2/4\sigma^2)}{4\pi\sigma^2}$$

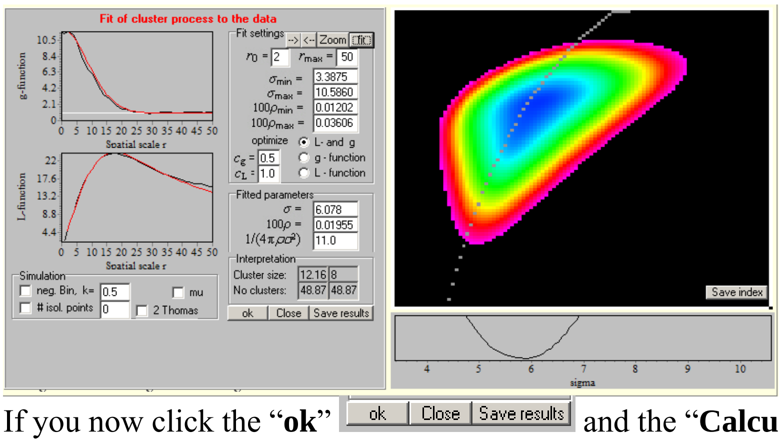
where the factor $f_k = (k + 1)/k$ yields $f_k = 1$ for the simple Thomas process using a Poisson distribution for the distribution p_S. In practical terms this means that clumping of points over the clusters does not change the functional form of the Thomas process, and that we can therefore find for each value of k a simple Thomas process (where p_S is a Poisson distribution) with exactly the same pair correlation (and K-) function. The parameter of this simple Thomas process that describes the number of clusters yields $\rho_\mathrm{S} = \rho f_k$ where ρ is the parameter of the generalized Thomas process. That means that clumping of the points over the clusters generates a pair correlation function that seems to have more clusters.

This also means that fitting with second order properties alone does not allow us to determine the parameter k of the generalized Thomas process. However, other summary functions of different nature such as the spherical contact distribution $H_\mathrm{s}(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor may allow us to approximate the value of k. Thus, we need to fit first with the pair correlation and the L-function and then simulate several generalized Thomas process with different values of the clumping parameter k, but adjust the number of clusters of the simulated generalized Thomas process in a way that the pair correlation function does not change (i.e., $\rho_\mathrm{S} = \rho f_k$). We then can check the fit of $H_\mathrm{s}(r)$ and $D(r)$ for different values of k and indirectly determine the value that fits $H_\mathrm{s}(r)$ and $D^k(r)$ best.

Example Book_Fig4_11e_k=1000.res (fit parameters σ, ρ, and k)

This example file was generated with a generalized Thomas process with parameters σ =6, $A\rho$ = 50, and k = 1 (f_k = 2).

1. Execute *Programita*.
2. Highlight data file Book_Fig4_11e.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Cluster process**" in the window "Select a null model".
9. A window "Fit of cluster process to data" opens. Select in the section "Null models" at the bottom "**Univar. cluster**". Continue with the small **ok** button.
10. Fit the parameters σ, and ρ. First provide distances interval (r_0, $r_{\max}$) for the fit (i.e., r_0 = 2 $r_{\max}$ = 50) and then click the small "fit" button: --> <-- Zoom fit *Programita* now tests the entire parameter space indicated by ($\sigma_{\min}$, $\sigma_{\max}$) and ($\rho_{\min}$, $\rho_{\max}$) with default values.
11. By clicking "zoom" and then "fit", *Programita* repeats the fit, but only for the neighborhood of the best fitting parameter combination. In this way you can improve the fit. As expected, you obtain a good fit (σ =6.1, $A\rho$ = 49 in the example below).



Fit of cluster process to the data

Fit settings --> <-- Zoom fit

r_0 = 2 $r_{\max}$ = 50

$\sigma_{\min}$ = 3.3875

$\sigma_{\max}$ = 10.5860

$100\rho_{\min}$ = 0.01202

$100\rho_{\max}$ = 0.03606

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 6.078

100ρ = 0.01955

$1/(4\pi\rho\sigma^2)$ 11.0

Interpretation

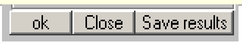
Cluster size: 12.16 8

No clusters: 48.87 48.87

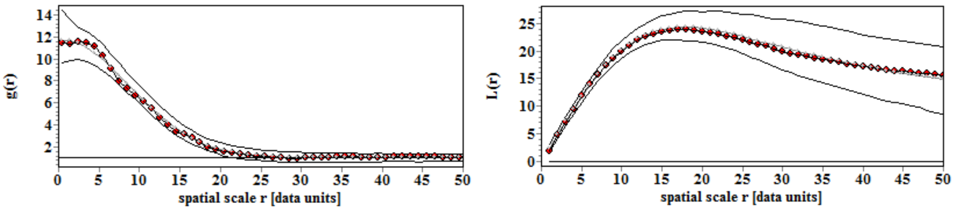
Simulation

□ neg. Bin, k= 0.5 □ mu

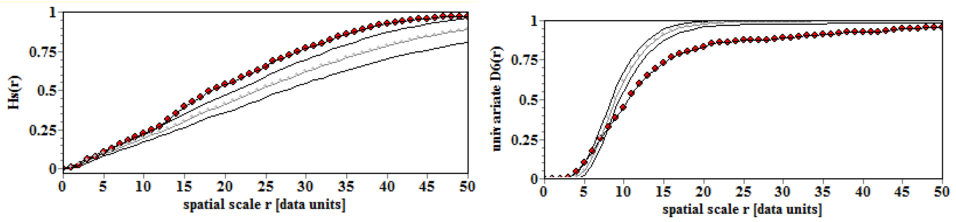
□ # isol. points 0 □ 2 Thomas

If you now click the "**ok**"  ok Close Save results and the "**Calculate Index**" you simulate the simple Thomas process (i.e., $k = \infty$).

12. As expected, the pair correlation and L- function are perfectly fitted:

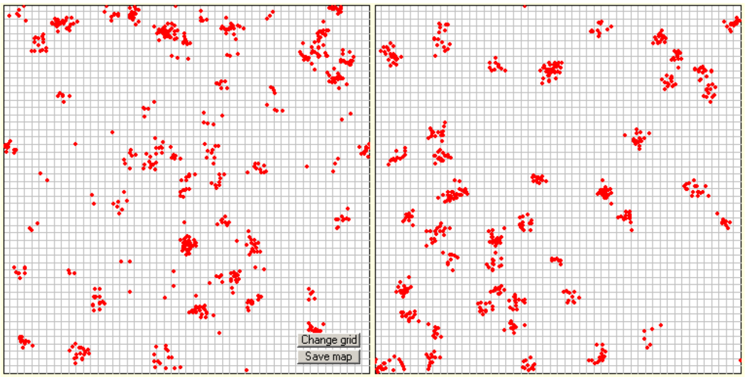


but the spherical contact distribution $H_s(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor (here the 6th neighbor) not:



The spherical contact distribution is underestimated which means that the gaps in the simulated pattern are too large and the distribution function of the distances to the 6th neighbor is overestimated which means that the 6th neighbors in the simulated patterns are in general too close.

When looking at the observed pattern and a realization of the simple Thomas process we detect only subtle differences, the observed pattern has somewhat more scattered points and the simulated pattern has more well delineated clusters with larger gaps:

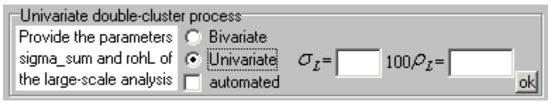


Automated fit of simple Thomas process

The simple Thomas process can also be fitted in an automated way by using the procedure developed for the Thomas process with two critical scales of clustering. Steps 1-8 are the same as before, then:

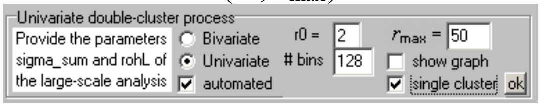
11. A window "Fit of cluster process to data" opens. Select in the section "Null models" at the bottom "**Univar. double-cluster**".

12. Select "Univariate" and click "automated"



Univariate double-cluster process
Provide the parameters sigma_sum and rohL of the large-scale analysis
○ Bivariate
● Univariate
□ automated
σ_I =
$100\rho_I$ =
ok

13. Enable "single cluster" and click the small "Ok" button. You can also change the distance interval (r0, $r_{\max}$)

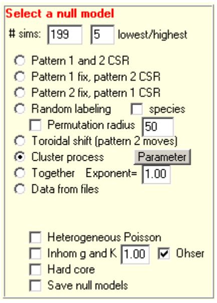


Univariate double-cluster process
Provide the parameters sigma_sum and rohL of the large-scale analysis
○ Bivariate
● Univariate
☑ automated
r0 = 2
# bins 128
$r_{\max}$ = 50
□ show graph
☑ single cluster
ok

14. *Programita* then fits the simple Thomas process over the interval (r0, $r_{\max}$) to the data.

Example Book_Fig4_11e_k=10.res (fit parameters σ, ρ, and k)

This example continues the above example Book_Fig4_11e_k=1000.res, but now simulates a generalized Thomas process with parameter k = 10.



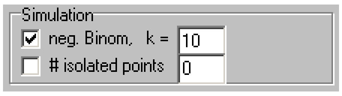
Select a null model

# sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
● Cluster process Parameter
○ Together Exponent= 1.00
○ Data from files

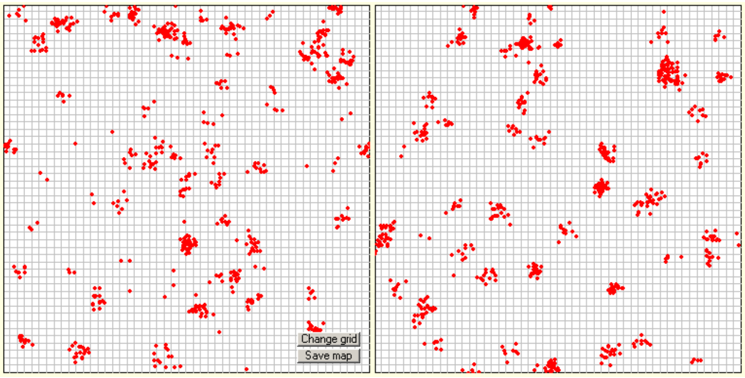
□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

1. The simulation of the simple Thomas process is terminated. To access the menu of the Thomas process click the button "**Parameters**" in the window "Select a null model".
2. To select a value of k = 10 that corresponds to f_k = 1.1, click the check box "**neg. Binom**" and write 10 in the corresponding text box:

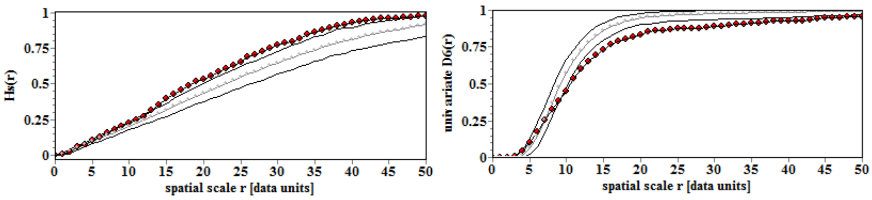


Simulation
☑ neg. Binom, k = 10
□ # isolated points 0

3. If you now click the "**ok**" and the "**Calculate Index**" you simulate the generalized Thomas process with k = 10 that fits the observed pair correlation and L-function. The simulated patterns now resemble the observed one better but there are still too large gaps:

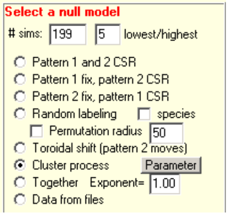


and the spherical contact distribution $H_s(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor (here the 6th neighbor) are only slightly better fitted than before:



Example Book_Fig4_11e_k=1.res (fit parameters *σ*, *ρ*, and *k*)

This example continues the above example Book_Fig4_11e_k=1000.res, but now simulates a generalized Thomas process with parameter $k = 1$ (the parameter used to generate the pattern).

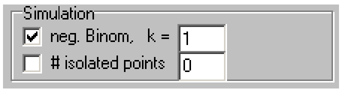


Select a null model

# sims: 199 5 lowest/highest

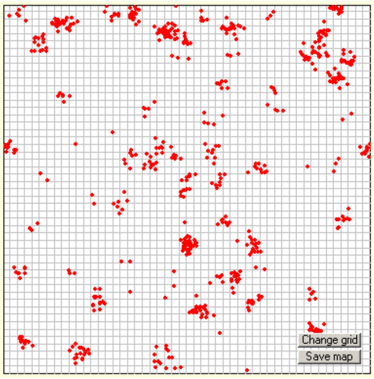
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
● Cluster process Parameter
○ Together Exponent= 1.00
○ Data from files

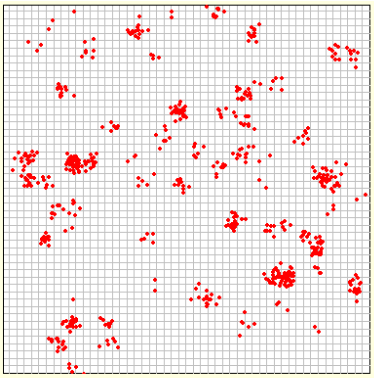
1. The simulation of the simple Thomas process is terminated. To access the menu of the Thomas process click the button "**Parameters**" in the window "Select a null model".
2. To select a value of $k = 1$ that corresponds to $f_k = 2$, click the check box "**neg. Binom**" and write 1 in the corresponding text box:



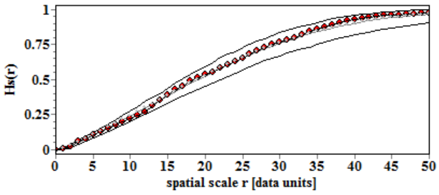
Simulation
☑ neg. Binom, k = 1
□ # isolated points 0

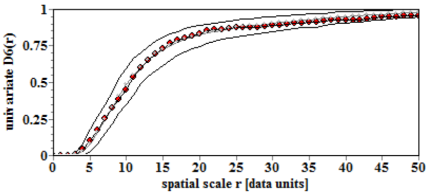
3. If you now click the "**ok**" and the "**Calculate Index**" you simulate the generalized Thomas process with $k = 1$ that fits the observed pair correlation and L-function. The simulated patterns now resemble the observed one very well:





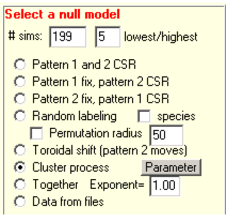
and the spherical contact distribution $H_s(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor (here the 6th neighbor) are also well fitted:



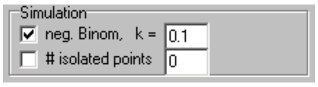


Example Book_Fig4_11e_k=01.res (fit parameters *σ*, *ρ*, and *k*)

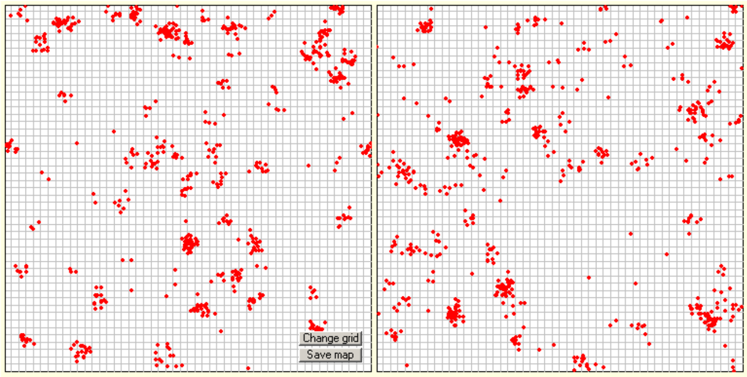
This example continues the above example Book_Fig4_11e_k=1000.res, but now simulates a generalized Thomas process with parameter $k = 0.1$.


Select a null model
# sims: 199 5 lowest/highest
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
● Cluster process
○ Together Exponent= 1.00
○ Data from files

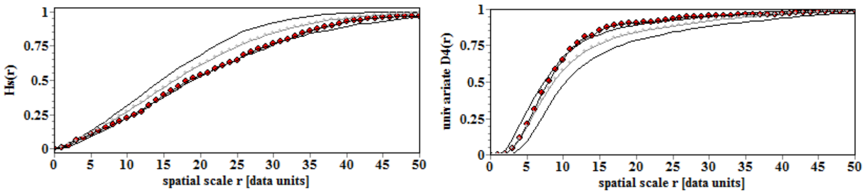
1. The simulation of the simple Thomas process is terminated. To access the menu of the Thomas process click the button “**Parameters**” in the window “Select a null model”.
2. To select a value of $k = 0.1$ that corresponds to $f_k = 11$, click check box “**neg. Binom**” and write 1 in the corresponding text box:


Simulation
☑ neg. Binom, k = 0.1
□ # isolated points 0

3. If you now click the “**ok**” and the “**Calculate Index**” you simulate the generalized Thomas process with $k = 0.1$ that fits the observed pair correlation and L-function. The simulated patterns now show a somewhat to dispersed pattern with too much scattered points:



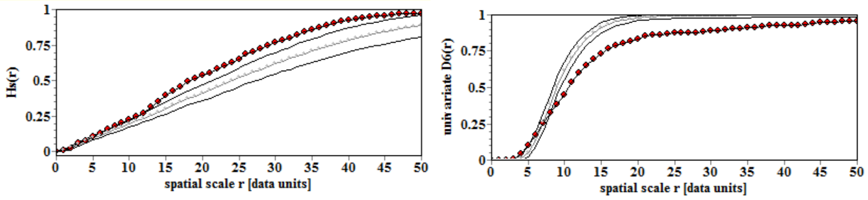
and the spherical contact distribution $H_s(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor (here the 4th neighbor) are not well fitted:

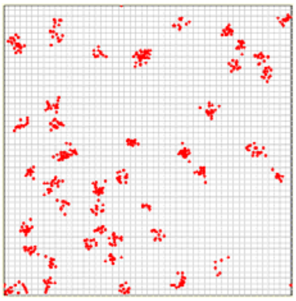


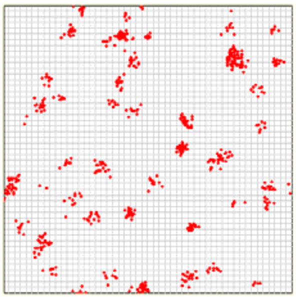
the simulated pattern now has too small gaps and the 4th neighbor is too far.

Summarizing the fits with the $D^k(r)$ we see that the spherical contact distribution $H_s(r)$ and the distribution function $D^k(r)$ of the distances to the *k*th neighbor (i.e., the pointwise simulation envelopes) change systematically with changing clumping parameter *k*:

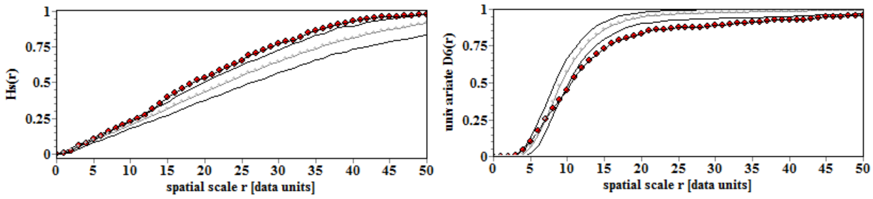
$k = \infty$:

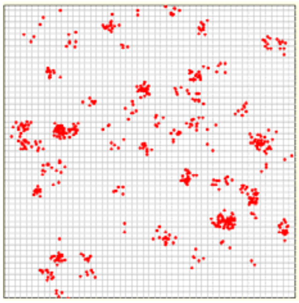




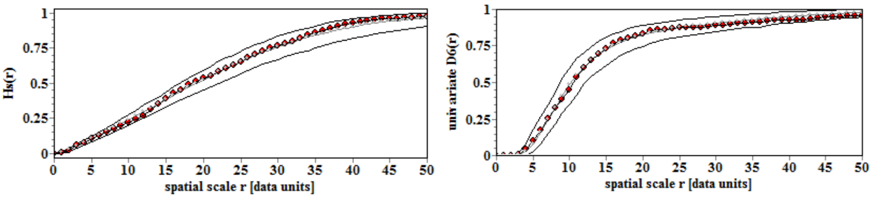


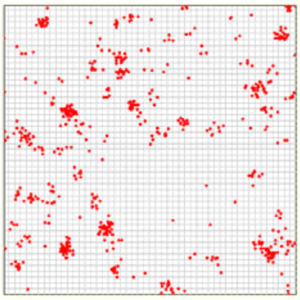
$k = 10$:



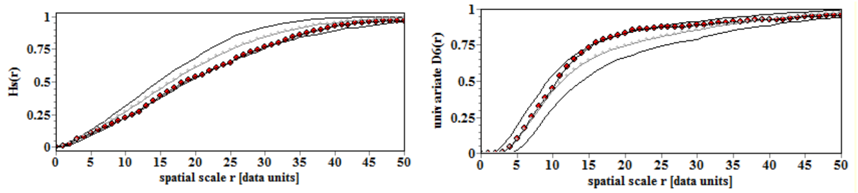


$k = 1$:





$k = 0.1$:



3.3.4 Superposition of CSR and a simple Thomas process

This example illustrates a second generalization of the simple Thomas process with one scale of clustering that allows description of a clustered pattern with more “isolated” points than expected by the simple Thomas process.

The first option to accomplish this was using the negative Binominal distribution p_S to describe the number of points S per cluster. Clearly, if k is relatively small (e.g., $k = 1$ or 0.1) there will be much more clusters with only one point (i.e., isolated points) than expected by the simple Thomas process (see e.g., insets in Figures 4.11b, f, and j in Wiegand and Moloney 2014). For example, the generalized Thomas process with $k = 0.1$ yields 550 clusters, 422 are empty (i.e., $S = 0$) and 42 have only one point (= 32% of all clusters with at least one point) while the corresponding simple Thomas process does not show isolated points (see Fig. 4.11b).

The second option is an independent superposition of a simple Thomas process with a CSR pattern (for details see sections 3.3 and 3.3.5 “Examples of the Superposition of a Thomas Process with a Random Pattern” in Wiegand and Moloney 2014). The nice feature of the Thomas process is the superposition with CSR does not change the functional form of the analytical solution of the pair correlation function (and the K-function) of the Thomas process. We obtain:

$$g(r,\rho,\sigma) = 1 + \frac{c^2}{\rho}\frac{\exp(-r^2/4\sigma^2)}{4\pi\sigma^2}$$

where the parameter c is the proportion of points of the pattern belonging to the Thomas process. In practical terms this means that superposition with CSR does not change the functional form of the Thomas process, and that we can therefore find for each value of c a simple Thomas process (where $c = 1$) with exactly the same pair correlation (and K-) function. The parameter of this simple Thomas process that describes the number of clusters yields $\rho_S = \rho/c^2$ where ρ is the parameter of the generalized Thomas process. That means that superposition with CSR generates a pair correlation function that seems to have more clusters.

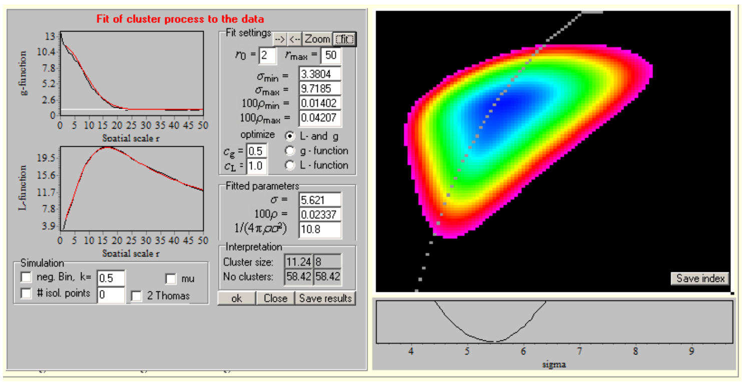
This also means that fitting with second order properties alone does not allow us to determine the proportion (1- c) of isolated CSR points. However, other summary functions of different nature such as the spherical contact distribution $H_S(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor may allow us to approximate the value of c. Thus, we need to fit first with the pair correlation and the L-function to obtain estimates of σ and ρ and then simulate several generalized Thomas processes with different proportions 1- c of random points, but adjust the number of clusters of the simulated generalized Thomas processes in a way that the pair correlation function does not change (i.e., $\rho_S = \rho/c^2$). We then can check the fit of $H_S(r)$ and $D(r)$ for different values of c and indirectly determine the value that fits $H_S(r)$ and $D^k(r)$ best.

Note that you cannot simulate an independent superposition of a random pattern with a generalized Thomas process with a negative Binominal distribution. Such a process is not implemented in *Programita* and it would be difficult to determine the parameters with any confidence.

Example Fig4_11CSR0.res (fit parameters σ, ρ, and c)

This example file was generated with a generalized Thomas process with parameters σ =6, $A\rho$ = 50, and k = 1 (c = 0.84; 100 isolated points).

1. Execute *Programita*.
2. Highlight data file Book_Fig4_11csr100.dat in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Cluster process**" in the window "Select a null model".
9. A window "Fit of cluster process to data" opens. Select in the section "Null models" at the bottom "**Univar. cluster**". Continue with the small **ok** button.
10. Fit the parameters σ, and ρ by clicking the "fit" button and if needed "Zoom" and "fit". As expected, you obtain a good fit (σ =5.4, $A\rho$ = 47 in the example below).



Fit of cluster process to the data

Fit settings
r_0 = 2 r_{max} = 50
σ_{min} = 3.3804
σ_{max} = 9.7185
$100\rho_{min}$ = 0.01402
$100\rho_{max}$ = 0.04207
optimize ● L- and g ○ g - function ○ L - function
c_g = 0.5
c_L = 1.0

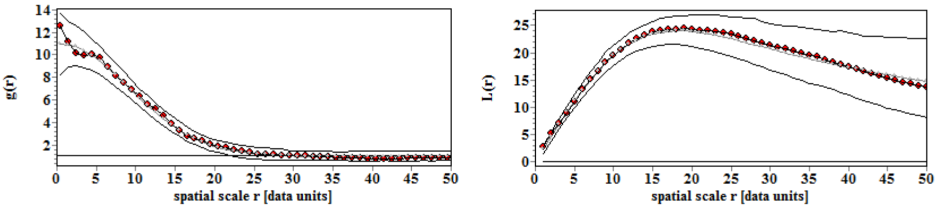
Fitted parameters
σ = 5.621
100ρ = 0.02337
$1/(4\pi\rho\sigma^2)$ = 10.8

Interpretation
Cluster size: 11.24 8
No clusters: 58.42 58.42

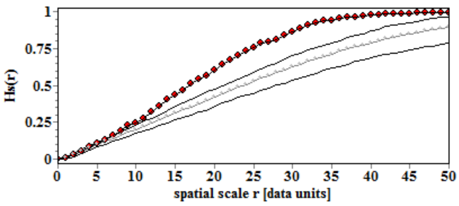
Simulation
□ neg. Bin, k= 0.5   □ mu
□ # isol. points 0   □ 2 Thomas

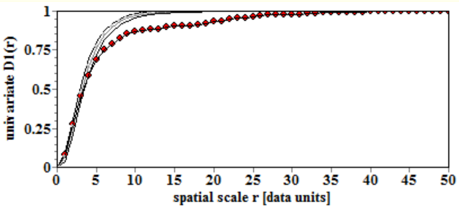
If you now click the "**ok**" and the "**Calculate Index**" you simulate the simple Thomas process (i.e., $k = \infty$, c = 1).

11. As expected, the pair correlation and L- function are perfectly fitted:



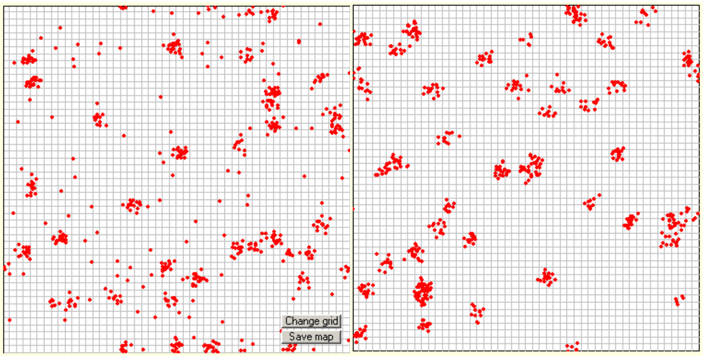
but the spherical contact distribution $H_s(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor (here the 1^{th} neighbor) not:





The spherical contact distribution is underestimated which means that the gaps in the simulated pattern are too large (clearly the interspersed random points are missing) and the distribution function of the distances to the nearest neighbor is overestimated which means that the nearest neighbors in the simulated patterns are in general too close. Clearly, the isolated points of the CSR component have their nearest neighbor not within a cluster but usually much farther away.

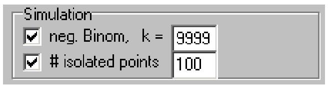
When looking at the observed pattern and a realization of the simple Thomas process we detect only subtle differences, the observed pattern has somewhat more isolated points:



Superposition of CSR and a simple Thomas process. Example *Book_Fig4_11CSR100.res*

This example continues the above example Fig4_11CSR0.res, but now simulates a generalized Thomas process with the parameter c = 0.84 (i.e., 100 isolated points) that were used to generate it.

1. The simulation of the simple Thomas process is terminated. To access the menu of the Thomas process click the button "**Parameters**" in the window "Select a null model".
2. To select 100 random points click the checkbox "**neg. Binom**" and "**# isolated points**" and write 100 in the corresponding text box:



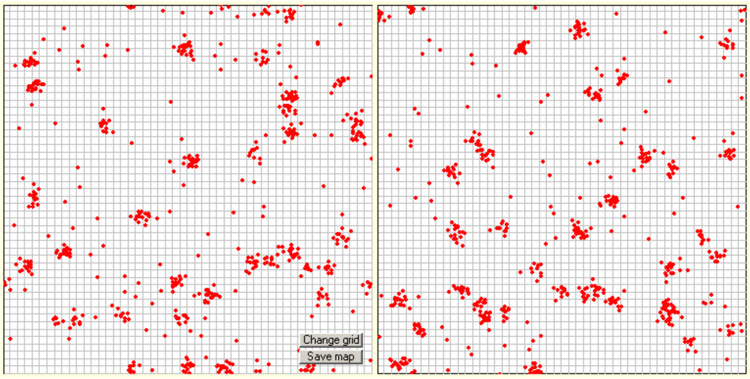
Simulation

☑ neg. Binom, k = 9999

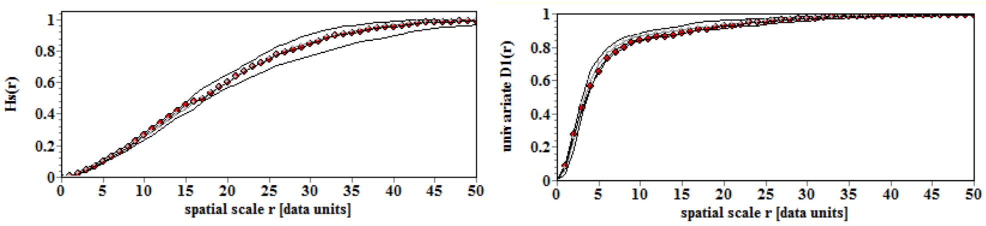
☑ # isolated points 100

A value of k = 9999 appears automatically to represent a Poisson distribution for p_S. If you would select a different value for k it will be reset to 9999 since this more complex combination of processes is not implemented in *Programita*.

3. If you now click the "**ok**" and the "**Calculate Index**" you simulate the simple Thomas process with 526 points, independently superposed with 100 random points that fits the observed pair correlation and L-function. The simulated patterns now resemble the observed one well:



and the spherical contact distribution $H_s(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor (here the 1th neighbor) fitted well:



3.3.5 Simple bivariate parent-offspring Thomas process

This example illustrates use of a Thomas process where the cluster centers are known. This cluster process was first presented in Jacquemyn et al. (2007) and Wiegand et al. (2007). In this case we have two patterns, pattern 1 are the cluster centers and pattern 2 the pattern that is assumed to follow a simple Thomas process.

Programita offers two point process models to analyze this data type:

- In the first case the bivariate second-order statistics between the points (pattern 2) and the cluster centers (pattern 1) is fitted.
- In the second case the standard Thomas process is used to fit the univariate pattern 2, but the simulation of the point process uses the known cluster centers (pattern 1).

The bivariate pair correlation functions of the bivariate pattern yields (equation 4.13 in Wiegand and Moloney 2014):

$$g_{12}(r,\sigma_2,\lambda_1) = 1 + \frac{1}{\lambda_1}\frac{\exp(-r^2/2\sigma_2^2)}{2\pi\sigma_2^2}$$

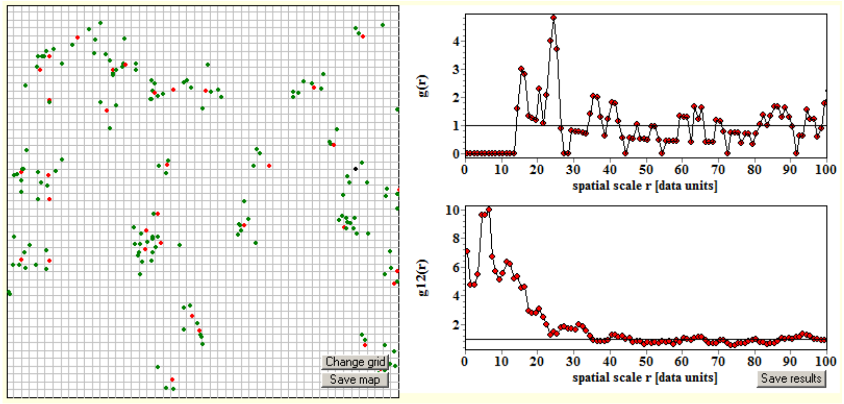
where $\lambda_1 = \rho_2$ is the intensity of pattern 1 (i.e., the cluster centers).

Programita fits the parameters λ_1 and σ_2 of this point process and then simulates the univariate Thomas process with known parents.

Example Book_Fig4_13_bi.res

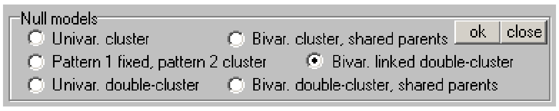
This example file was generated with a simple Thomas process with 157 points and parameters σ_2 = 13.3 and $A\ \lambda_1$ = 35 random clusters.

1. Execute *Programita*.
2. Highlight data file Book_Fig4_13.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Click button "change" below to set maximal distance *r* to be analyzed. Insert 100 in small box that opens and then the small **ok** button.
6. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
7. Press button "**Calculate Index**"
8. *Programita* then shows the pattern, the univariate pair correlation function of the cluster centers and the bivariate pair correlation function of the points around their parents:



The pair correlation function of pattern 1, which is not of interest here, is somewhat rugged because pattern 1 has few points (i.e., 34 clusters).

9. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
10. Select "**Cluster process**" in the window "Select a null model".
11. A window "Fit of cluster process to data" opens. Select in the section "Null models" at the bottom the button "**Bivar. linked double-cluster**".

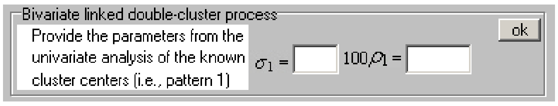

Null models

○ Univar. cluster ○ Bivar. cluster, shared parents

○ Pattern 1 fixed, pattern 2 cluster ● Bivar. linked double-cluster

○ Univar. double-cluster ○ Bivar. double-cluster, shared parents

Now this window appears that asks you to provide the parameters from the univariate analysis of pattern 1 (i.e., the cluster centers):


Bivariate linked double-cluster process

Provide the parameters from the univariate analysis of the known cluster centers (i.e., pattern 1)

σ_1 = 100ρ_1 =

This is because the implementation of *Programita* uses here the equation for the more general point process where the cluster centers follow a simple Thomas process. See section 4.1.4.4 "Bivariate Parent–Offspring Thomas Processes" in Wiegand and Moloney (2014). The equation used for fitting is equation 4.14:

$$g_{12}(r,\rho_1,\sigma_1,\rho_2,\sigma_2) = \frac{1}{\rho_2}h(r,\sigma_2)+\frac{1}{\rho_1}h(r,\sqrt{2\sigma_1^2+\sigma_2^2})$$

$$= 1+\frac{1}{\rho_2}\frac{\exp(-r^2/(2\sigma_2^2))}{2\pi\sigma_2^2}+\frac{1}{\rho_1}\frac{\exp(-r^2/(4\sigma_1^2+2\sigma_2^2))}{4\sigma_1^2+2\sigma_2^2}$$

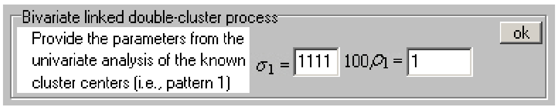
Because pattern 1 is in the example pattern "Book_Fig4_13.dat" a random pattern, you can recover the equation of the more simpler point process where the cluster centers following CSR, i.e.,

$$g_{12}(r,\sigma_2,\lambda_1) = 1 + \frac{1}{\lambda_1}\frac{\exp(-r^2/2\sigma_2^2)}{2\pi\sigma_2^2}$$

by selecting parameters σ_1 and ρ_1 that correspond to CSR for pattern 1, e.g., σ_1 = 1111 and $100\rho_1 = 1$. In this case the second term

$$\frac{1}{\rho_1} h(r, \sqrt{2\sigma_1^2 + \sigma_2^2})$$

approximates zero. Thus, insert σ_1 = 1111 and $100\rho_1 = 1$

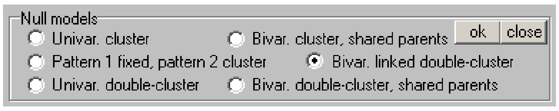


Bivariate linked double-cluster process

Provide the parameters from the univariate analysis of the known cluster centers (i.e., pattern 1)

σ_1 = 1111 $100\rho_1$ = 1

and click the small "**ok**" button and then again the small "**ok**" button in the next window:



Null models

○ Univar. cluster

○ Pattern 1 fixed, pattern 2 cluster

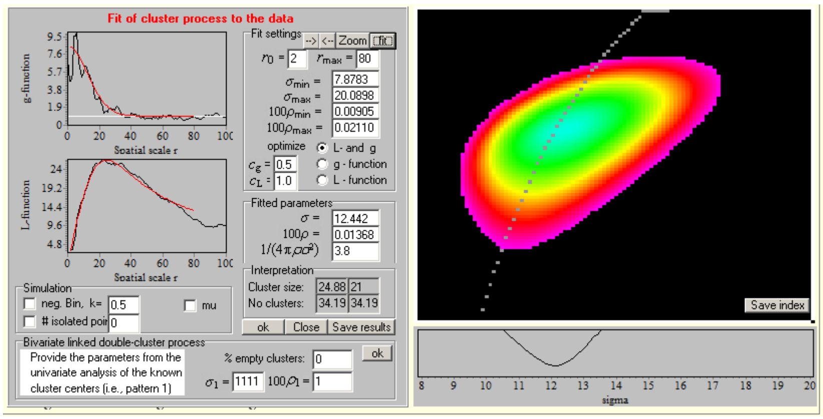
○ Univar. double-cluster

○ Bivar. cluster, shared parents

● Bivar. linked double-cluster

○ Bivar. double-cluster, shared parents

12. Fit the parameters σ, and ρ over the distance 1 to 80:



Fit of cluster process to the data

Fit settings

r_0 = 2 r_{max} = 80

σ_{min} = 7.8783

σ_{max} = 20.0898

$100\rho_{min}$ = 0.00905

$100\rho_{max}$ = 0.02110

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 12.442

100ρ = 0.01368

$1/(4\pi\rho\sigma^2)$ 3.8

Interpretation

Cluster size: 24.88 21

No clusters: 34.19 34.19

Simulation

□ neg. Bin, k= 0.5 □ mu

□ # isolated poir 0

Bivariate linked double-cluster process

Provide the parameters from the univariate analysis of the known cluster centers (i.e., pattern 1)

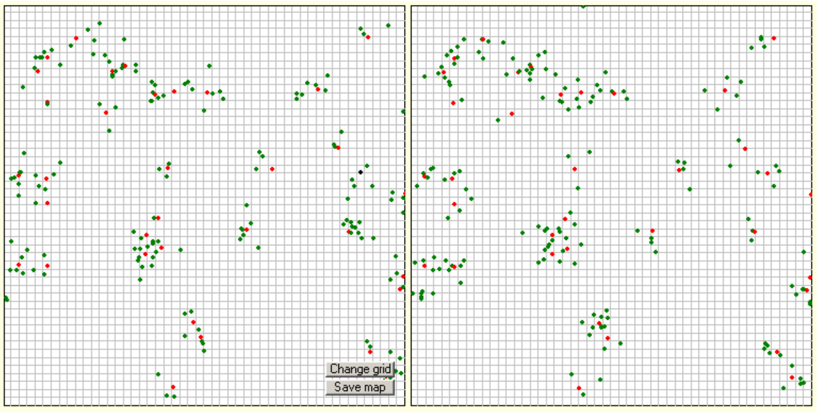
% empty clusters: 0

σ_1 = 1111 $100\rho_1$ = 1

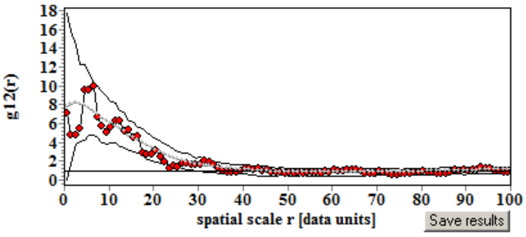
As indicated by the *L*-function, the fit is good and even the fitted number of cluster centers (34.2) coincides well with the known number (i.e., 34).

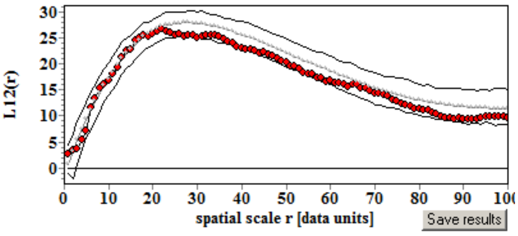
If the fitted number of cluster centers is larger than the number of points of pattern 1 (here 34), a warning appears and the known number of points of pattern 1 is used instead of the fitted parameter $A\rho_2$ for the number of cluster centers. If the fitted number of clusters is smaller than the known number of points of pattern 1 a reduced number of cluster centers is randomly selected among the points of pattern 1.

Click the small "**ok**" buttons and the "**Calculate Index**". *Programita* now simulates the pattern 2 of the fitted point process using the locations of pattern 1 as cluster centers:

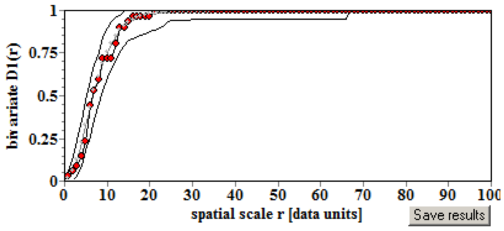


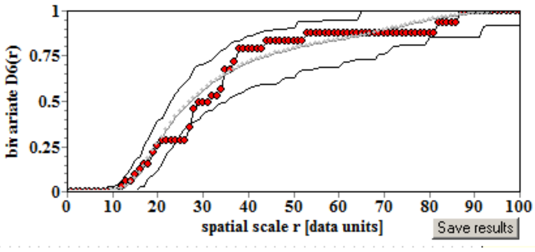
13. The pair correlation and *L*- function are well fitted:



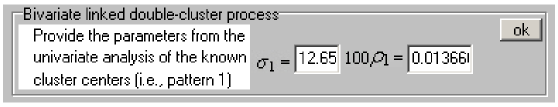


and the distribution functions $D^k(r)$ of the distances to the kth neighbors as well (here the 1^{th} and 6^{th} neighbor):





Note that fitting the bivariate parent-offspring Thomas process where the known cluster centers (i.e., pattern 1) are themselves clustered follows exactly the same procedure. The only difference is that you need to provide in step 11 the parameters fitted to the clustered pattern 1. Test for example with the data of file Book_Fig4_14.dat:



Bivariate linked double-cluster process

Provide the parameters from the univariate analysis of the known cluster centers (i.e., pattern 1)

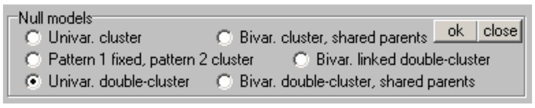
σ_1 = 12.65   100ρ_1 = 0.01366(

Note also that the bivariate parent-offspring Thomas processes with known parents are somewhat sensitive to the assumption that the parents follow CSR or a simple Thomas process. If this assumption is not well met, you may better use the equivalent Cox process presented in example Book_Fig4_18.res below which makes no assumptions on the pattern of the cluster centers.

Example Book_Fig4_13_uni.res

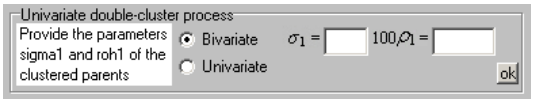
This example analyzes the same data set as example Book_Fig4_13_bi.res, but with a different method. In this case pattern 2 is fitted to a simple Thomas process and then in the simulation of the fitted point process the points of pattern 1 are used as cluster centers.

1. Execute *Programita*.
2. Highlight data file Book_Fig4_13.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Cluster process**" in the window "Select a null model".
9. A window "Fit of cluster process to data" opens. Select in the section "Null models" at the bottom the button "**Univar. double cluster**".



Null models
○ Univar. cluster ○ Bivar. cluster, shared parents
○ Pattern 1 fixed, pattern 2 cluster ○ Bivar. linked double-cluster
● Univar. double-cluster ○ Bivar. double-cluster, shared parents

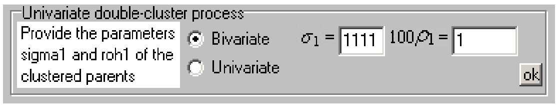
Now the below window appears. Select radio button "Bivariate" because you use the points of pattern 1 as cluster centers. Provide the parameters from the univariate analysis of pattern 1 (i.e., the cluster centers):



Univariate double-cluster process
Provide the parameters sigma1 and roh1 of the clustered parents
● Bivariate
○ Univariate
σ_1 =
$100\rho_1$ =

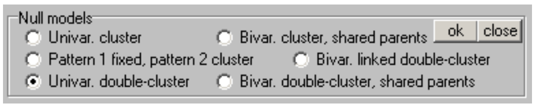
Again, this is because the implementation of *Programita* uses here the equation for the more general point process where the cluster centers follow a simple Thomas process. See section 4.1.4.5 "Generalized Thomas Process with Two Nested Scales of Clustering" in Wiegand and Moloney (2014). The equation used for fitting is equation 4.17.

Because pattern 1 is in the example pattern "Book_Fig4_13.dat" a random pattern, you can, in the same way as in the previous example, recover the equation of the simple Thomas process where the cluster centers following CSR by selecting parameters σ_1 and ρ_1 that correspond to CSR for pattern 1, e.g., σ_1 = 1111 and $100\rho_1$ = 1. Thus, insert σ_1 = 1111 and $100\rho_1$ = 1



Univariate double-cluster process
Provide the parameters sigma1 and roh1 of the clustered parents
● Bivariate
○ Univariate
σ_1 = 1111
$100\rho_1$ = 1

and click the small "**ok**" button and then again the small "**ok**" button in the next window.



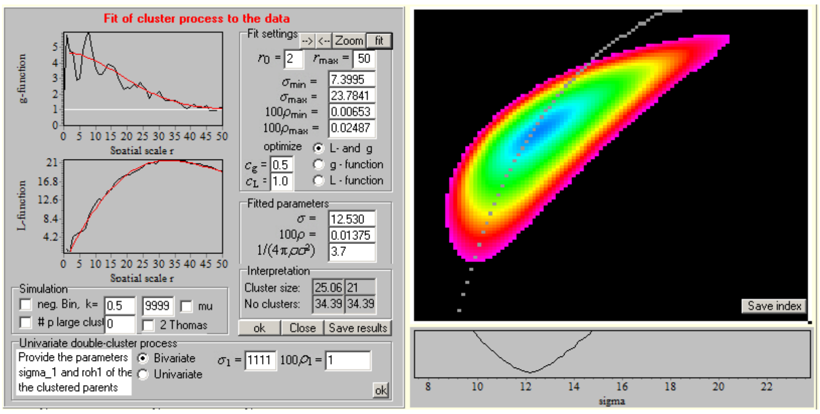
Null models

○ Univar. cluster ○ Bivar. cluster, shared parents

○ Pattern 1 fixed, pattern 2 cluster ○ Bivar. linked double-cluster

● Univar. double-cluster ○ Bivar. double-cluster, shared parents

10. Fit the parameters σ, and ρ over the distance 1 to 50 by clicking the small button "fit":



Fit of cluster process to the data

Fit settings

r_0 = 2 r_{max} = 50

σ_{min} = 7.3995

σ_{max} = 23.7841

$100\rho_{min}$ = 0.00653

$100\rho_{max}$ = 0.02487

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 12.530

100ρ = 0.01375

$1/(4\pi\rho\sigma^2)$ 3.7

Interpretation

Cluster size: 25.06 21

No clusters: 34.39 34.39

Simulation

☐ neg. Bin, k= 0.5 9999 ☐ mu

☐ # p large clust 0 ☐ 2 Thomas

Univariate double-cluster process

Provide the parameters sigma_1 and roh1 of the the clustered parents

● Bivariate ○ Univariate

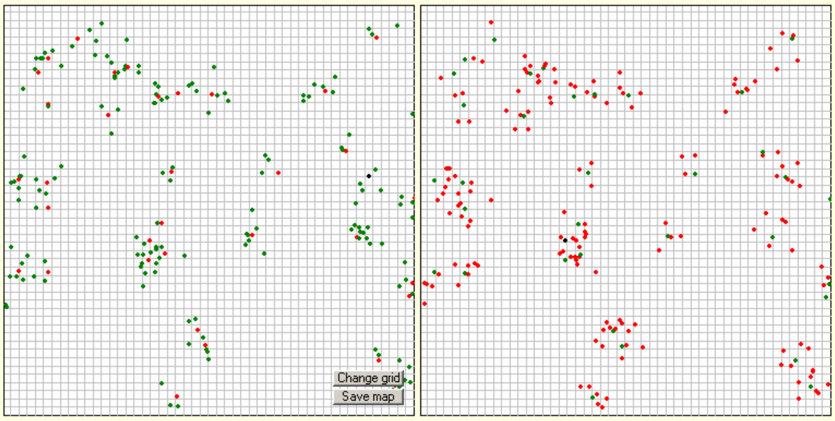
σ_1 = 1111 $100\rho_1$ = 1

With clicking "zoom" you can restrict the parameter space and obtain a better fit.

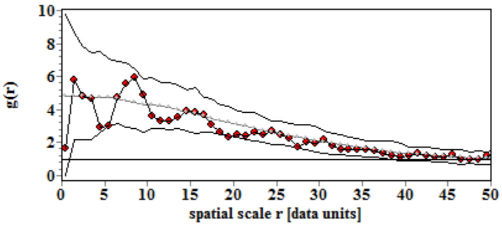
As indicated by the *L*-function, the fit is good and even the fitted number of cluster centers (34.4) coincides well with the known number (i.e., 34).

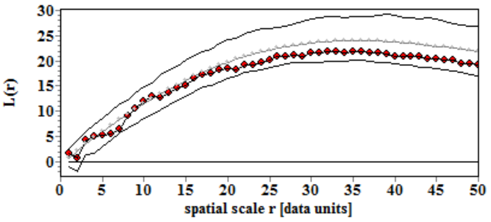
If the fitted number of cluster centers is larger than the number of points of pattern 1 (here 34), a warning appears and the known number of points of pattern 1 is used instead of the fitted parameter $A\rho_2$ for the number of cluster centers. If the fitted number of clusters is smaller than the known number of points of pattern 1 a reduced number of cluster centers is randomly selected among the points of pattern 1.

11. Click the small "**ok**" buttons and the "**Calculate Index**". *Programita* now simulates the pattern 2 of the fitted point process using the locations of pattern 1 as cluster centers (note that the colors are exchanged in the simulated pattern):

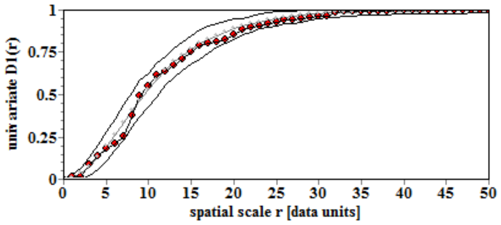


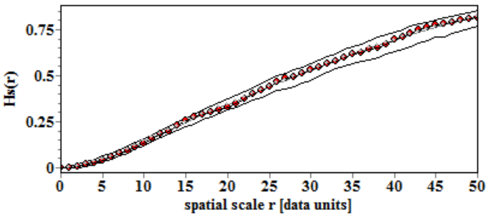
The pair correlation and *L*- function are well fitted:



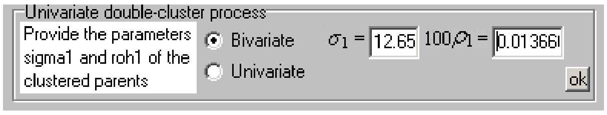


and the distribution functions $D^k(r)$ of the distances to the kth neighbors (here the 1th neighbor) and the spherical contact distribution as well:





Note that fitting the bivariate parent-offspring Thomas process where the known cluster centers (i.e., pattern 1) are themselves clustered follows exactly the same procedure. The only difference is that you need to provide in step 12 the parameters fitted to the clustered pattern 1:



Univariate double-cluster process

Provide the parameters sigma1 and roh1 of the clustered parents

● Bivariate

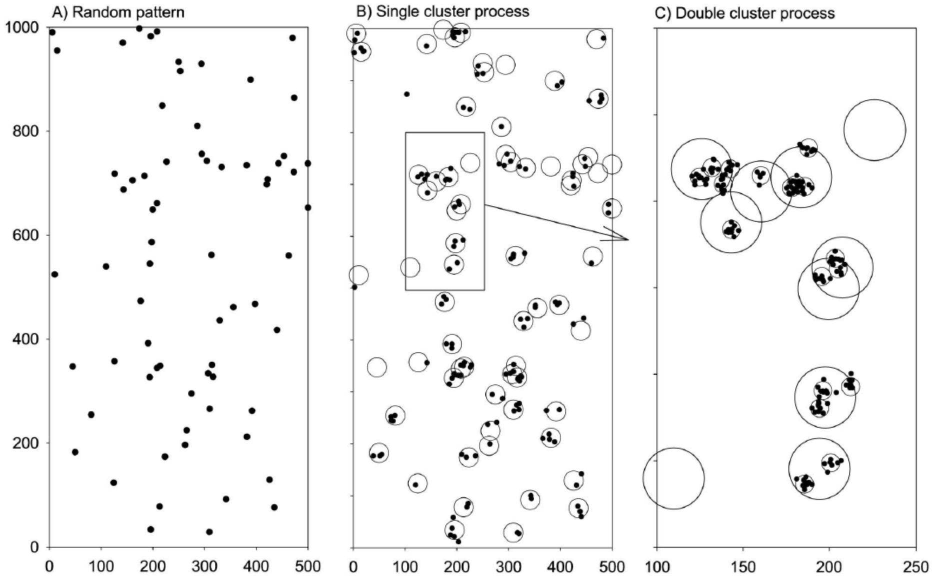
○ Univariate

σ_1 = 12.65 100ρ_1 = 0.01366

Test for example with the data file Book_Fig4_14.dat

3.3.6 Thomas process with two nested scales of clustering

This point process is a generalization of the (generalized) simple Thomas process where the pattern of cluster centers does not follow CSR (as in the generalized simple Thomas process), but can itself be a (generalized) simple Thomas process. This leads to clusters inside clusters and a nested cluster structure as shown below:



Details on this point process model can be found in section 4.1.4.5 “Generalized Thomas Process with Two Nested Scales of Clustering” in Wiegand and Moloney (2014).

The pair correlation function of this point process is shown in equation 4.17:

$$g(r,\rho_L,\sigma_L,\rho_S,\sigma_S) = 1 + \underbrace{\frac{f_{k_S}}{\rho_S} h(r,\sqrt{2}\sigma_S)}_{\text{small-scale clustering}} + \underbrace{\frac{f_{k_L}}{\rho_L} h(r,\sqrt{2\sigma_L^2 + 2\sigma_S^2})}_{\text{combined effect of small- and large-scale clustering}}$$

where $h(r,\sigma) = \dfrac{\exp(-0.5r^2/\sigma^2)}{2\pi\sigma^2}$. This point process has six free parameters:

- σ_S: the parameter that describes the size of the small clusters
- ρ_S: the intensity of the small clusters (governing the number of small clusters)
- $f_{kS} = (k_S + 1)/k_S$ where k_S is the parameter of the negative Binominal distribution that governs the distribution of points over the small clusters
- σ_L: the parameter that describes the size of the large clusters
- ρ_L: the intensity of the large clusters (governing the number of large clusters)
- $f_{kL} = (k_L + 1)/k_L$ where k_L is the parameter of the negative Binominal distribution that governs the distribution of the number of small clusters over the large clusters

The above equation for the pair correlation function of the double cluster process is composed of three summands. The first summand (1) is the contribution which remains if the two other summands disappears when the pattern is a CSR pattern. The second summand is only due to the small-scale clustering, and the third summand contains an interaction between small- and large-scale clustering, indicated by σ_{sum}.

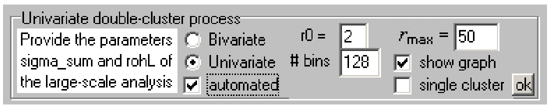
As we saw before when introducing the generalized simple Thomas process, the parameters k_{L} and k_{S} cannot be fit with the *g*- or *L*-function, but need to be indirectly determined by comparing the resulting fit of other summary functions such as the spherical contact distribution and the distribution function of the distances to the kth neighbor. Thus, we are left with the task to fit four unknown parameters to the second-order summary functions.

Programita uses for this a simple two-step procedure which is based on separation of scales. If the two critical scales of clustering are well separated, i.e., $\sigma_{\text{S}} << \sigma_{\text{L}}$

- we can first fit the parameters ρ_{L} and σ_{L} of the large scale clustering using a distance interval that starts with distance larger than $\approx 2\sigma_{\text{S}}$. In this case the second summand will be very small and because of $\sigma_{\text{S}} << \sigma_{\text{L}}$ the third summand will be dominated by σ_{L}. Thus, in this case we basically fit the generalized simple Thomas process to the data and obtain unbiased estimates for the parameters ρ_{L} and σ_{sum} of the large scale clustering.
- Second, we use the estimates of ρ_{L} and σ_{L} and fit the unknown parameters ρ_{S} and σ_{S} of the small scale clustering now using the entire distance interval. Finally, we use the estimate of σ_{S} to estimate σ_{L} using $\sigma_L^2 = \sigma_{sum}^2 - \sigma_S^2$. Note that the fitting procedure of *Programita* uses σ_{sum} in the third summand and estimates σ_{S} via the second summand.

Algorithm to fit univariate double cluster Thomas process

Programita uses an automated fitting algorithm based on the two-step idea outlined above. One essential auxiliary parameter of the two-step approach is the distance $r_{0\text{L}}$ where the small-scale clustering component just disappears. While this parameter can be relatively well adjusted by hand, the automatic procedure requires repeating the fit for different values of $r_{0\text{L}}$ and to select the one that produces the best overall fit.
Programita conducts the fit over distance interval (r_0, r_{max}) which you can select in the settings of the double cluster Thomas process.

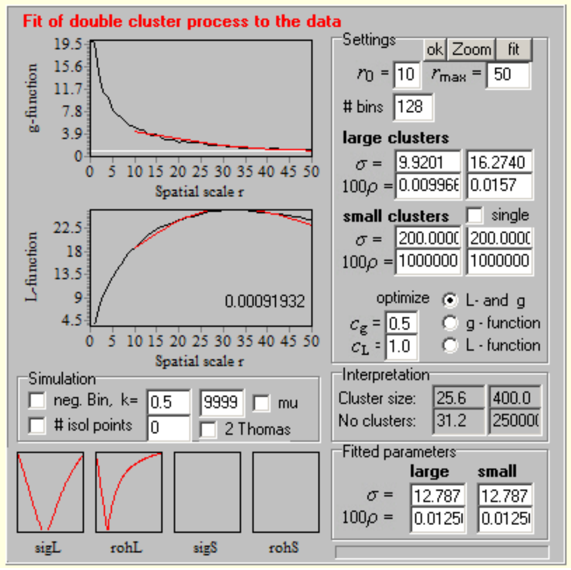


Univariate double-cluster process
Provide the parameters sigma_sum and rohL of the large-scale analysis
○ Bivariate
● Univariate
☑ automated
r0 = 2
# bins 128
r_{max} = 50
☑ show graph
□ single cluster

Programita varies the auxiliary parameter $r_{0\text{L}}$ over the interval (r_0, $r_{0,\text{max}}$) where $r_{0,\text{max}}$ = trunc(r_{max}/2.5). For example for $r_{\text{max}} = 50$, $r_{0\text{L}}$ is varied over the interval (r_0, 20) in steps of Δr = 1+trunc([$r_{0\text{max}}$–r_0]/10). For $r_0 = 2$ and $r_{\text{max}} = 50$ we obtain steps of $\Delta r = 2$.

Thomas process with two nested scales of clustering. Example *Book_Fig4_15b.res*

To fit the four parameters σ_L, ρ_L, σ_S, ρ_S for a given value of r_{0L} (subscripts L and S stand for large- and small-scale clustering, respectively), *Programita* fits first the two parameters ρ_L and σ_L of the large-scale clustering:



Fit of double cluster process to the data

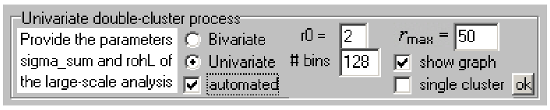
Settings
r_0 = 10 r_{max} = 50
# bins 128
large clusters
σ = 9.9201 16.2740
100ρ = 0.009966 0.0157
small clusters □ single
σ = 200.0000 200.0000
100ρ = 1000000 1000000
optimize ● L- and g
c_g = 0.5 ○ g - function
c_L = 1.0 ○ L - function

Simulation
□ neg. Bin, k= 0.5 9999 □ mu
□ # isol points 0 □ 2 Thomas

Interpretation
Cluster size: 25.6 400.0
No clusters: 31.2 250000

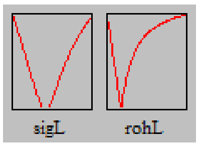
Fitted parameters
large **small**
σ = 12.787 12.787
100ρ = 0.0125 0.0125

To do this *Programita* samples the ρ_L - σ_L parameter space based on all parameter combinations of a grid. To do this the parameter interval of each parameter is divided per default into 128 bins (this setting can be changed).



Univariate double-cluster process
Provide the parameters sigma_sum and rohL of the large-scale analysis
○ Bivariate
● Univariate
☑ automated
r0 = 2
# bins 128
r_{max} = 50
☑ show graph
□ single cluster

Programita minimizes for the fit the discrepancy *error_Lg* between the observed and the theoretical *g*(*r*) and *L*(*r*) predicted by the double-cluster Thomas process over the interval (r_{0L}, r_{max}). In the example, r_{0L} = 10. Smaller *r* values are not considered in the first step of the fitting procedure.



The small plots show the minimal value of *error_Lg* for a given value of σ_L (taken over all values of ρ_L) (left) and for a given value of ρ_L (taken over all values of σ_L) (right). The minimum should be sharp and in the middle of the interval.

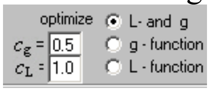
The discrepancy *error_Lg* is the geometric mean of the individual discrepancies *error_L* and *error_g* of the *L*- and *g*-function, respectively. For example, the *error_g* is basically the sum of squares of the difference between observed and theoretical *g*(*r*), by divided by the total sum of squares of the observed *g*(*r*):

$$error\_L = \sum_{r=r_0}^{r\max}[\hat{L}(r)^c - L(r,\sigma,\rho)^c]^2 \Big/ \sum_{r=r_0}^{r\max}[\hat{L}(r)^c]^2$$

$$error\_g = \sum_{r=r_0}^{r\max}[\hat{g}(r)^c - g(r,\sigma,\rho)^c]^2 \Big/ \sum_{r=r_0}^{r\max}[\hat{g}(r)^c]^2$$

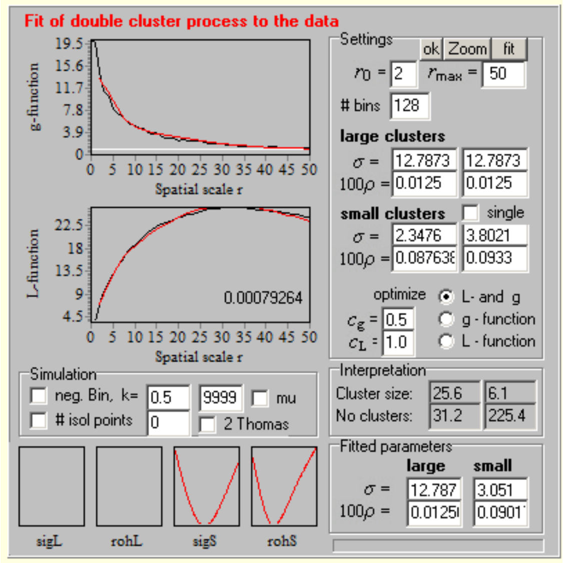
$$error\_Lg = \sqrt{error\_g * error\_L}$$

Additionally, to have more flexibility, the *L*- and *g*-function are power transformed with power *c*. If you like, you can base the fit also only on the pair correlation function or only on the *L*-function. However, it is recommended to base the fit on both second-order statistics because the *g*-function is more sensitive to small-scale clustering and the *L*-function to large-scale clustering. The corresponding settings can be changed here:



optimize ● L- and g
c_g = 0.5 ○ g - function
c_L = 1.0 ○ L - function

In a next step the parameters σ_L = 12.78 and ρ_L = 0.0125/100 of the large-scale clustering are kept fixed and the parameters σ_S and ρ_S of the small-scale clustering are fitted.


Fit of double cluster process to the data

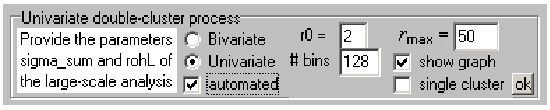
Settings ok Zoom fit
r_0 = 2 r_{max} = 50
# bins 128
large clusters
σ = 12.7873 12.7873
100ρ = 0.0125 0.0125
small clusters □ single
σ = 2.3476 3.8021
100ρ = 0.087638 0.0933
optimize ● L- and g ○ g - function ○ L - function
c_g = 0.5
c_L = 1.0

Interpretation
Cluster size: 25.6 6.1
No clusters: 31.2 225.4

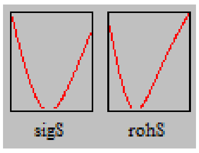
Fitted parameters
large small
σ = 12.787 3.051
100ρ = 0.01251 0.0901

Simulation
□ neg. Bin, k= 0.5 9999 □ mu
□ # isol points 0 □ 2 Thomas

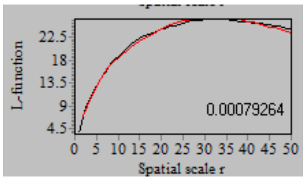
To do this *Programita* now samples the ρ_S - σ_S parameter space based on all parameter combinations of a grid. To do this the parameter interval of each parameter is divided per default into 128 bins (this setting can be changed).


Univariate double-cluster process
Provide the parameters sigma_sum and rohL of the large-scale analysis
○ Bivariate
● Univariate
☑ automated
r0 = 2
# bins 128
r_{max} = 50
☑ show graph
□ single cluster ok

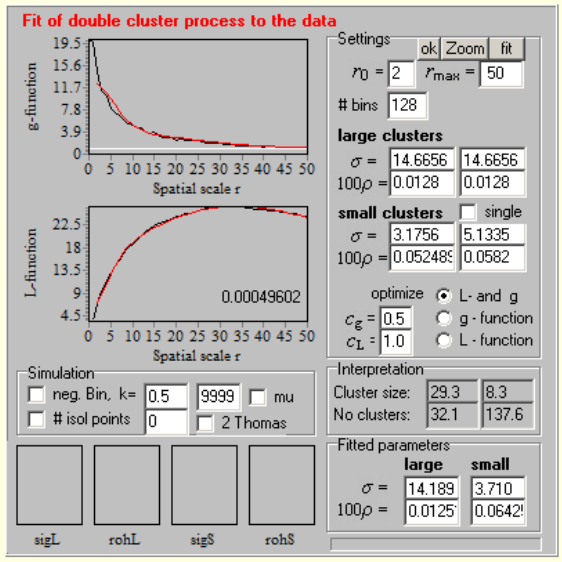
Programita minimizes for the fit again the discrepancy *error_Lg* between the observed and the theoretical $g(r)$ and $L(r)$ predicted by the double-cluster Thomas process, but now over the entire distance interval (r_0, r_{max}).



The small plots show the value of *error_Lg* for a given value of σ_S (taken over all values of ρ_S) (left) and for a given value of ρ_S (taken over all values of σ_S) (right). The minimum should be sharp and in the middle of the interval.



The examples shows a good fit with *error_Lg* = 0.000793. For each value of the auxiliary parameter r_{0L} *Programita* memorizes the smallest *error_Lg* and the corresponding parameters. Once all values of r_{0L} are tested, the final fit is given by those that show the smallest value of *error_Lg*.


Fit of double cluster process to the data

Settings ok Zoom fit
r_0 = 2 r_{max} = 50
# bins 128
large clusters
σ = 14.6656 14.6656
100ρ = 0.0128 0.0128
small clusters □ single
σ = 3.1756 5.1335
100ρ = 0.052489 0.0582
optimize ● L- and g ○ g - function ○ L - function
c_g = 0.5
c_L = 1.0

Interpretation
Cluster size: 29.3 8.3
No clusters: 32.1 137.6

Fitted parameters
large small
σ = 14.189 3.710
100ρ = 0.0125 0.0642

Simulation
□ neg. Bin, k= 0.5 9999 □ mu
□ # isol points 0 □ 2 Thomas

As shown on the left, *Programita* found a slightly better fit with *error_Lg* = 0.000496.

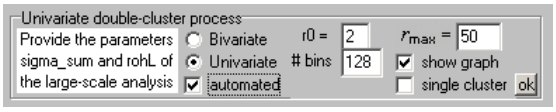
Note that the velocity of the fit depends on the square of the number of bins, but more bins allows for a more detailed fit. In general, 128 bins produce good fits.

Example Book_Fig4_15b.res (fit parameters σ_L, ρ_L, σ_S, ρ_S)

Now I explain the fitting procedure for the double-cluster Thomas process step-by-step. This example file is shown in Figure 4.15a in Wiegand and Moloney (2014). The data are the locations of small saplings of the species *Shorea congestiflora* from the Sinharaja plot in Sri Lanka. This data set was analyzed in detail in Wiegand et al. (2007a).

To analyze this data set we first fit the Thomas process with two nested scales of clustering to the data, following the automated two-step procedure outlined above, and then simulate this point process with different parameters f_{kL} and f_{kS} of the negative Binominal distribution.

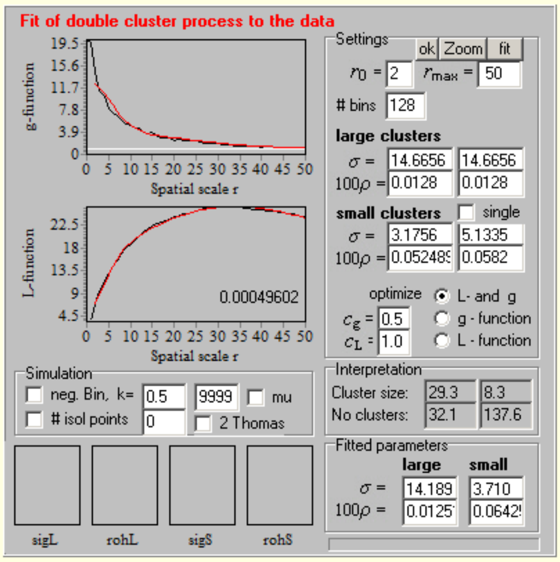
1. Execute *Programita*.
2. Highlight data file Book_Fig4_15a.dat you want to analyze in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
5. Press button “**Calculate Index**”
6. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
7. Select “**Cluster process**” in the window “Select a null model”.
8. A window “Fit of cluster process to data” opens. Select in the section “Null models” at the bottom “**Univar. cluster**”. Next enable “Univariate” and “automated”:



Univariate double-cluster process
Provide the parameters sigma_sum and rohL of the large-scale analysis
○ Bivariate
● Univariate
☑ automated
r0 = 2
# bins 128
r_{max} = 50
☑ show graph
□ single cluster
ok

Select also the distance interval (r_0, r_{max}) used for fitting. You can also change the number of bins used for each parameter in the σ-ρ parameter space. Note that the time requirement for the fit scales with (# bins)^2. Finally click the small “**ok**” button to start the automatic fitting procedure.

9. Once Programita has finished this task, this window is shown:



Fit of double cluster process to the data

Settings
ok | Zoom | fit
r_0 = 2 r_{max} = 50
# bins 128

large clusters
σ = 14.6656 14.6656
100ρ = 0.0128 0.0128

small clusters □ single
σ = 3.1756 5.1335
100ρ = 0.052489 0.0582

optimize
● L- and g
○ g - function
○ L - function
c_g = 0.5
c_L = 1.0

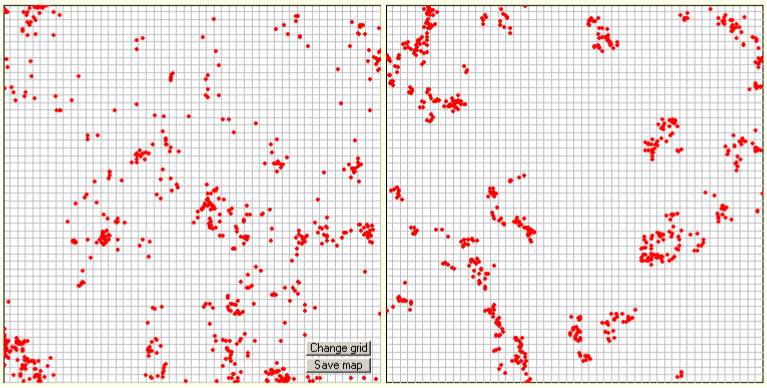
Simulation
□ neg. Bin, k= 0.5 9999 □ mu
□ # isol points 0 □ 2 Thomas

Interpretation
Cluster size: 29.3 8.3
No clusters: 32.1 137.6

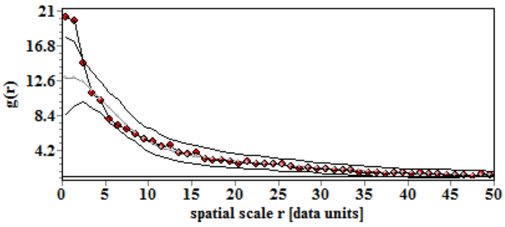
Fitted parameters
large small
σ = 14.189 3.710
100ρ = 0.0125 0.0642

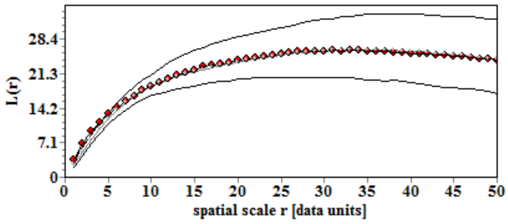
10. Finally, click the small **“ok”** button and then “**Calculate Index**” to simulate the fitted point process.

11. The simulated patterns are similar to the observed one, but there are not enough isolated points:

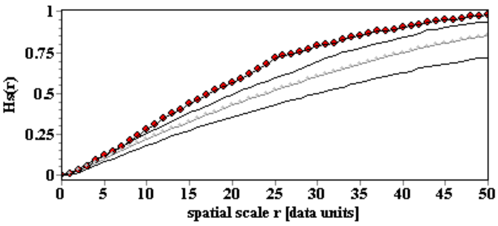


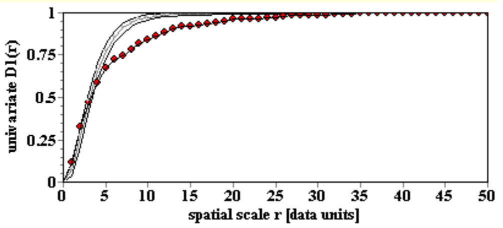
12. The pair correlation function is well fitted, but there is an additional clustering at very small distances < 3m which also causes some smaller departures in the *L*-function:





However, the spherical contact distribution is clearly underestimated (i.e., the gaps are too large in the simulated patterns) and the distribution function of the distances to the nearest neighbor are overestimated at distances > 5m which indicates that the data have more isolated points than the data generated by the cluster process:

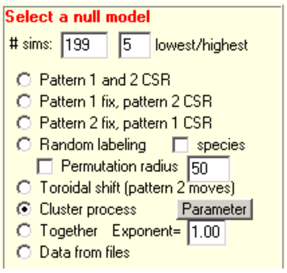




Examples Book_Fig4_15f.res (fit parameters f_{kL}, f_{kS})

This example continues the previous example Book_Fig4_15b.res by using a negative Binomial distribution to generate a clumped distribution of the number of saplings over the small clusters. We select as in Figure 4.15f $k_L = \infty$ ($f_{kL} = 1$) and $k_S = 0.1$ ($f_{kS} = 11$).

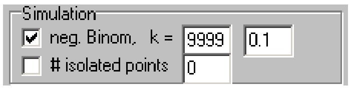
13. To access the parameter fitting window click “Parameters” at the Select a null model window, click “**neg. Binom**” at the “simulation” window, insert $k = 9999$ for the large scale clustering, $k = 0.1$ for the small scale clustering, and then click the small **“ok”** button:



Select a null model

# sims: 199 5 lowest/highest

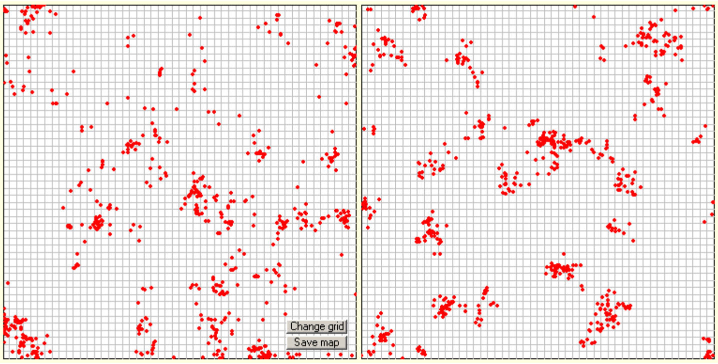
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
● Cluster process Parameter
○ Together Exponent= 1.00
○ Data from files



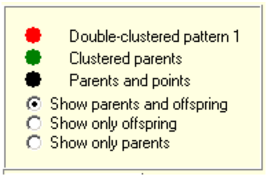
Simulation

☑ neg. Binom. k = 9999 0.1
□ # isolated points 0

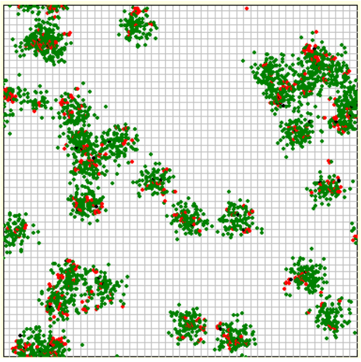
14. After pressing the “**Calculate Index**” button *Programita* simulates the generalized cluster process with two nested scales of clustering. Comparison between data and simulated patterns shows that the simulated pattern still has too large gaps:



If you select in the small window below the simulated pattern “Show parents and offspring” you can see additionally to the simulated pattern (red points) the cluster centers (of the small-scale clustering) as green points:

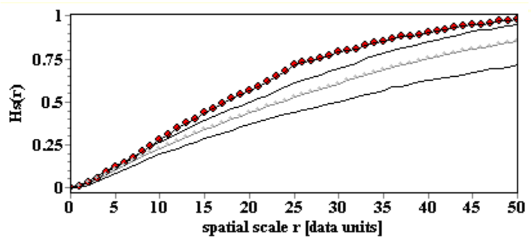


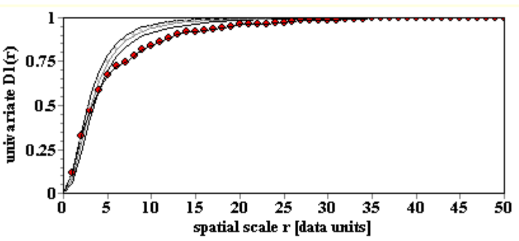
● Double-clustered pattern 1
● Clustered parents
● Parents and points
● Show parents and offspring
○ Show only offspring
○ Show only parents



15. Clearly, because of $k_{\mathrm{S}} = 0.1$ and $f_{k\mathrm{S}} = 11$ you have 11 times as much cluster centers as in the example above with $f_{k\mathrm{S}} = 1$.

The fit of the spherical contact distribution does not improve but the fit of the distribution function of the distances to the nearest neighbor somewhat improves, thus the important feature of the data is probably not the distribution of the number of points over the small clusters:



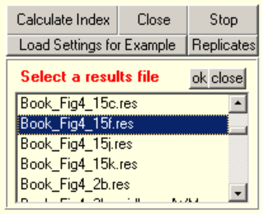


Examples Book_Fig4_15j.res (fit parameters f_{kL}, f_{kS})

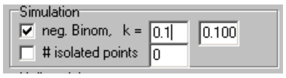
You can easily verify that the point process analogous to Book_Fig4_15f.res with parameters $k_L = 0.1$ ($f_{kL} = 11$) and $k_S = \infty$ ($f_{kS} = 1$) does also not much improve the fit of the spherical contact distribution and the distribution function of the distances to the nearest neighbor.

This example therefore continues the previous example Book_Fig4_15f.res by using a negative Binomial distribution to generate a clumped distribution of both, the number of small clusters over the large clusters and the number of saplings over the small clusters. We select as in Figure 4.15j parameters $k_L = 0.1$ ($f_{kL} = 11$) and $k_S = 0.1$ ($f_{kS} = 11$).

1. To access the parameter settings of example Book_Fig4_15f.res click "**Load Settings for Example**", highlight file Book_Fig4_15f.res, and click the small **"ok"** button.



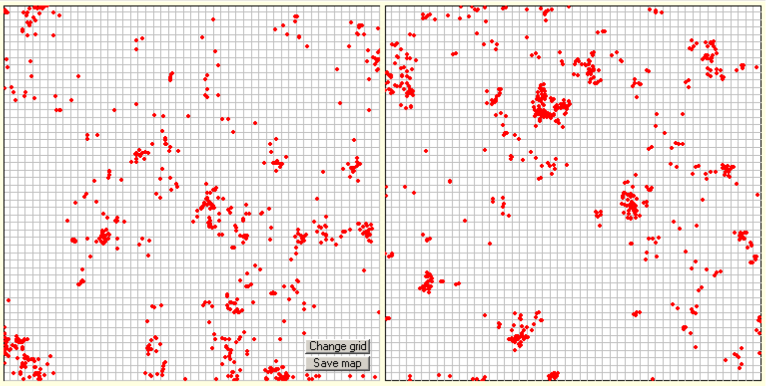
Book_Fig4_15c.res
Book_Fig4_15f.res
Book_Fig4_15j.res
Book_Fig4_15k.res
Book_Fig4_2b.res



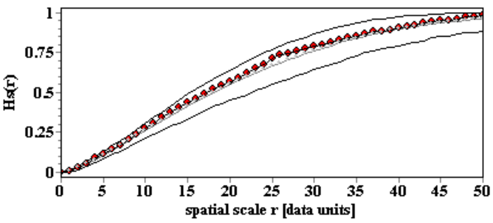
☑ neg. Binom, k = 0.1 0.100
□ # isolated points 0

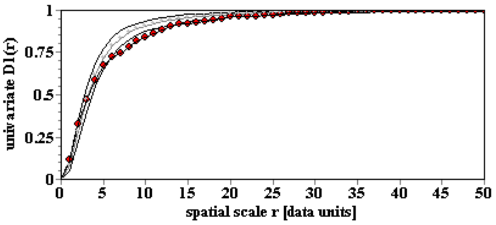
Now change in the "simulation" window the parameters k to insert $k = 0.1$ for the large scale clustering and $k = 0.1$ for the small scale clustering, and then click the small "**ok**" button.

2. After pressing the "**Calculate Index**" button *Programita* simulates the generalized cluster process with two nested scales of clustering. Comparison between data and simulated patterns shows that the simulated pattern looks now very similar to the observed pattern:



The fit of the $H_s(r)$ is now perfect, and that of the $D_L(r)$ substantially improved. Note also that the GoF test of the $D^k(r)$ was not significant for $k > 3$:





3.3.7 Superposition of CSR and a double cluster process

In the previous examples Book_Fig4_15b.res, Book_Fig4_15f.res, and Book_Fig4_15j.res we generalized the double-cluster Thomas process to produce a pattern with the same second-order structure but more isolated points. This was done by manipulating the distribution of the number of points over the clusters and by increasing the number of clusters.

As already shown for the simple Thomas process in example Fig4_11CSR100.res, we can also generate a pattern with the same second-order structure but more isolated points if we independently superimpose a simple double cluster Thomas process (i.e., $f_{k\mathrm{L}} = 1$ and $f_{k\mathrm{S}} = 1$) with a CSR pattern (for details see section 4.1.4.6 “Independent Superposition of CSR within a Double Cluster Process”). A nice feature of the Thomas processes is that superposition with CSR does not change the functional form of the analytical solution of the pair correlation function (and the *K*-function). We obtain:

$$g(r,\sigma_L,\rho_L,\sigma_S,\rho_S) = 1 + \underbrace{\frac{p_C^2}{\rho_S} h(r,\sqrt{2}\sigma_S)}_{\substack{\text{small-scale}\\ \text{clustering}}} + \underbrace{\frac{p_C^2}{\rho_L} h(r,\sqrt{2\sigma_L^2 + 2\sigma_S^2})}_{\substack{\text{combined effect of small-}\\ \text{and large-scale clustering}}}$$

where $h(r,\sigma) = \dfrac{\exp(-0.5r^2/\sigma^2)}{2\pi\sigma^2}$

and the parameter p_C is the proportion of points of the pattern belonging to the double-cluster Thomas process.

In practical terms this means that superposition with CSR does not change the functional form of the Thomas process, and that we can therefore find for each value of p_C a double cluster Thomas process (where $p_C = 1$) with exactly the same pair correlation (and *K*-) function. The parameter of this double cluster Thomas process that describes the number of clusters yields $\rho_\mathrm{S} = \rho/p_C^2$ where ρ is the parameter fitted to the superposition process. That means that superposition with CSR generates a pair correlation function that seems to have more clusters. Comparison with the pair correlation function of the generalized double cluster process

$$g(r,\rho_L,\sigma_L,\rho_S,\sigma_S) = 1 + \underbrace{\frac{f_{k_S}}{\rho_S} h(r,\sqrt{2}\sigma_S)}_{\substack{\text{small-scale}\\ \text{clustering}}} + \underbrace{\frac{f_{k_L}}{\rho_L} h(r,\sqrt{\sigma_L^2 + \sigma_S^2})}_{\substack{\text{combined effect of small-}\\ \text{and large-scale clustering}}}$$

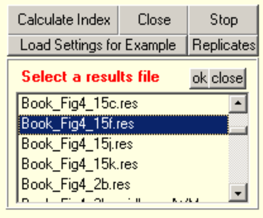
shows that the parameter p_C^2 of the superposition process plays the same role as the parameters $f_{k\mathrm{L}}$ and $f_{k\mathrm{S}}$ in the generalized double cluster Thomas process.

As before, this means that fitting with second order properties alone does not allow us to determine the proportion (1- p_C) of isolated CSR points. However, other summary functions of different nature such as the spherical contact distribution $H_\mathrm{S}(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor may allow us to approximate the value of p_C. The procedure for this is exactly the same as for the generalized double cluster Thomas process described above.

Examples Book_Fig4_16e.res (superposition with random pattern)

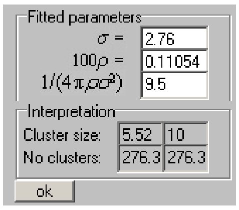
This example continues the previous example Book_Fig4_15c.res by independently superimposing a simple double cluster Thomas process (i.e., $f_{kL} = 1$ and $f_{kS} = 1$) with CSR.

1. To access the parameter settings of example Book_Fig4_15c.res click "**Load Settings for Example**", highlight file Book_Fig4_15c.res, and click the small "**ok**" button. Then click the small "**ok**" button at the window "Fitted parameters" and the button "Parameters" at the Select a null model window:



Select a results file

Book_Fig4_15c.res
Book_Fig4_15f.res
Book_Fig4_15j.res
Book_Fig4_15k.res
Book_Fig4_2b.res

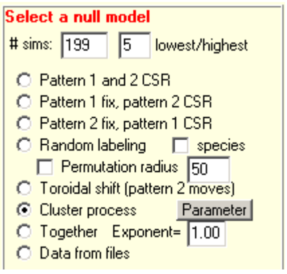


Fitted parameters

σ = 2.76
100ρ = 0.11054
$1/(4\pi\rho\sigma^2)$ 9.5

Interpretation

Cluster size: 5.52 10
No clusters: 276.3 276.3

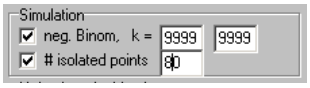


Select a null model

sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
● Cluster process
○ Together Exponent= 1.00
○ Data from files

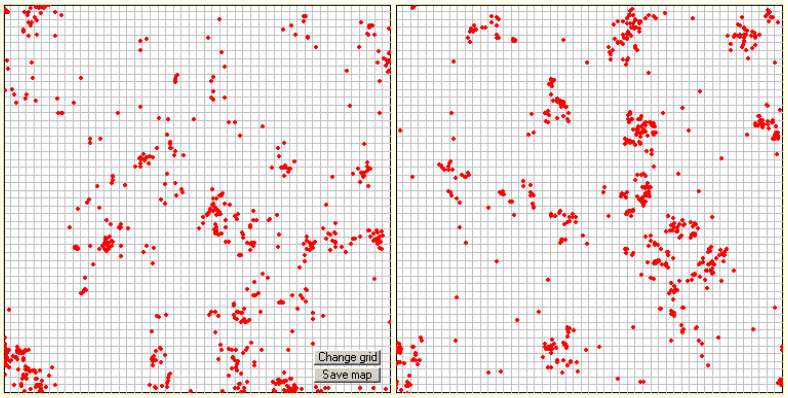
2. Now click "# isolated points", insert the number of points of the pattern that belong to the CSR pattern (80 in the example),


Simulation

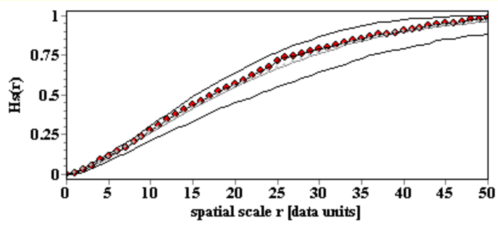
☑ neg. Binom, k = 9999 9999
☑ # isolated points 80

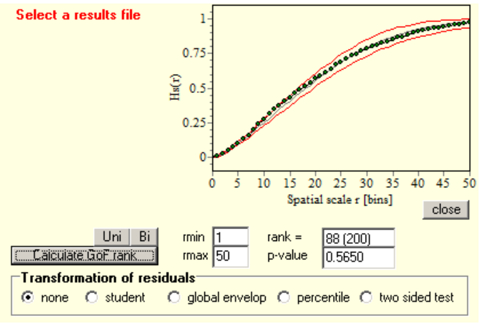
and finally click the small "**ok"** button at the window "Fitted parameters".

3. After pressing the "**Calculate Index**" button *Programita* simulates the superposition point process. Comparison between data and simulated patterns shows that the simulated pattern looks very similar to the observed pattern:



The fit of the $H_s(r)$ is very good, the GoF test over distance interval 1-50m yields a P-value of 0.36:



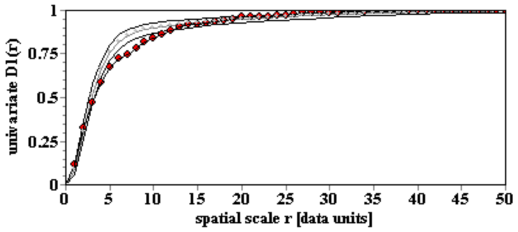


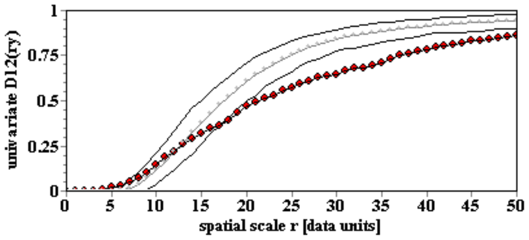
Select a results file

rmin 1
rmax 50
rank = 88 (200)
p-value 0.5650

Transformation of residuals
● none ○ student ○ global envelop ○ percentile ○ two sided test

However, while the distribution function of the distances to the nearest neighbor is relatively close to the data, the $D_k(r)$ fails for higher neighborhood ranks k, for example for $k = 12$:





That means that the internal structure of the clusters is not well represented by the superposition process. This indicates that the generalized double cluster Thomas process presented above in example Book_Fig4_15f.res is a more likely model for this data set.

3.3.8 Superposition of two Thomas processes

The previous example Book_Fig4_15b.res showed a Thomas process with two scales of clustering where the clusters where nested; i.e., small clusters where located inside large clusters. However, this is not the only possibility to generate univariate patterns with two scales of clustering. Stoyan and Stoyan (1996) proposed a point process resulting from the independent superposition of two simple Thomas processes. The additional parameter of this point process is the proportion p_L of the points that belong to the Thomas process with large-scale clustering. The pair correlation function for this point process yields:

$$g(r,\sigma_L^s,\rho_L^s,\sigma_S^s,\rho_S^s) = 1 + \underbrace{\frac{(1-p_L)^2}{\rho_S^s} h(r,\sqrt{2}\,\sigma_S^s}_{\text{small-scale clustering}} + \underbrace{\frac{p_L^2}{\rho_L^s} h(r,\sqrt{\sigma_L^{s2}+\sigma_L^{s2}})}_{\text{large-scale clustering}}$$

where $h(r,\sigma) = \dfrac{\exp(-0.5r^2/\sigma^2)}{2\pi\sigma^2}$.

For example, if both component processes have the same number of points (i.e., $p_\mathrm{L} = 0.5$), the contribution of clustering $h(r, 2^{0.5}\sigma)/\rho$ of each component is only one quarter of that of the original process. Thus, clustering of the superposition process is substantially reduced compared with the clustering of the two component processes. (If we have also $\sigma_\mathrm{L} = \sigma_\mathrm{S}$ we obtain, as expected, a process with only one scale of clustering, but with the double number of clusters).

For comparison, the pair correlation function nested double cluster process yields:

$$g(r,\rho_L,\sigma_L,\rho_S,\sigma_S) = 1 + \underbrace{\frac{f_{k_S}}{\rho_S} h(r,\sqrt{2}\sigma_S)}_{\text{small-scale clustering}} + \underbrace{\frac{f_{k_L}}{\rho_L} h(r,\sqrt{\sigma_L^2+\sigma_S^2})}_{\text{combined effect of small- and large-scale clustering}}$$

Thus, the pair correlation function of the superposition of two simple Thomas processes has the same structure as that of the nested double cluster process. We can therefore fit the parameters ρ_L, σ_L, ρ_S, and σ_S by using the procedure of the nested double cluster process. However, comparing the two equations suggests that small adjustments in the parameters are needed. If we fit with the procedure for the nested double cluster process we obtain parameters ρ_L, σ_L, ρ_S, and σ_S, but to describe the superposition process we have parameters $\rho^\mathrm{s}{}_\mathrm{L}$, $\sigma^\mathrm{s}{}_\mathrm{L}$, $\rho^\mathrm{s}{}_\mathrm{S}$, and $\sigma^\mathrm{s}{}_\mathrm{S}$. The parameter σ_S does not change (i.e., $\sigma_\mathrm{S}{}^\mathrm{s} = \sigma_\mathrm{S}$), but we need to adjust the parameter $\sigma_\mathrm{L}{}^\mathrm{s}$ that determines the size of the large clusters to

$$\sigma_L^s = \sqrt{\sigma_L^2 - \sigma_S^2}$$

and the parameters ρ_L and ρ_S that determine the number of cluster centers must be transformed to $\rho_1^s = \rho_1\, p_1^2$ and $\rho_2^s = \rho_1\,(1-p_1^2)$. *Programita* adjusts this automatically, but the *.res file outputs the parameters ρ_L, σ_L, ρ_S, and σ_S.

Thus, the two components Thomas processes of the superposition process have fewer cluster centers than the corresponding nested process. Or in other words, to yield the same pair correlation function for the nested and the superposition double cluster process, the superposition process must have substantially less clusters than the corresponding nested process (i.e., $\rho^{s}_{L} = \rho_{L}\, p^{2}_{L}$ and $\rho^{s}_{S} = \rho_{S}\,(1 - p^{2}_{L})$.

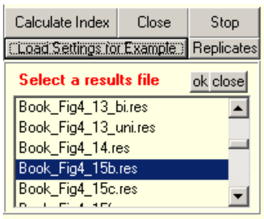
One consequence of this is that in cases where the proportion p_L of the points that belong to the Thomas process with large-scale clustering is small or large, we may have cases with (formally) less than one small or large cluster, respectively. Thus, one has to check if the parameters of the process make sense. In cases where one component process has less than one cluster *Programita* gives a warning and you can change the parameter p_L to yield exactly one cluster. However, with one cluster the point process is not well represented; it should have several clusters to make sense.

As before, fitting with second order properties alone does not allow us to determine the proportion p_L of the points that belong to the Thomas process with large-scale clustering. However, other summary functions of different nature such as the spherical contact distribution $H_S(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor may allow us to approximate the value of p_L.

Examples Fig4_15super.res (superposition of two Thomas processes)

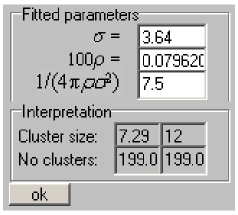
This example continues the previous example Book_Fig4_15b.res by fitting the point process that independently superimposes two simple Thomas processes to the data.

1. To access the parameter settings of example Book_Fig4_15b.res click "**Load Settings for Example**", highlight file Book_Fig4_15b.res, and click the small "**ok**" button. Then click the small "**ok"** button at the window "Fitted parameters" and the button "Parameters" at the Select a null model window:



Select a results file

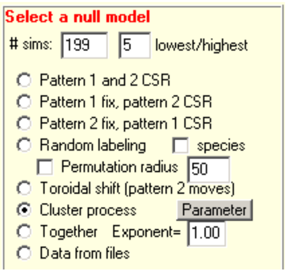
Book_Fig4_13_bi.res
Book_Fig4_13_uni.res
Book_Fig4_14.res
Book_Fig4_15b.res
Book_Fig4_15c.res


Fitted parameters

σ = 3.64
100ρ = 0.079620
1/(4$\pi\rho\sigma^2$) 7.5

Interpretation

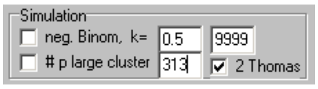
Cluster size: 7.29 12
No clusters: 199.0 199.0


Select a null model

sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
● Cluster process
○ Together Exponent= 1.00
○ Data from files

2. Now click "**2 Thomas**", and insert the number of points of the large-scale Thomas process (313 in the example yielding a proportion p_L = 0.5),

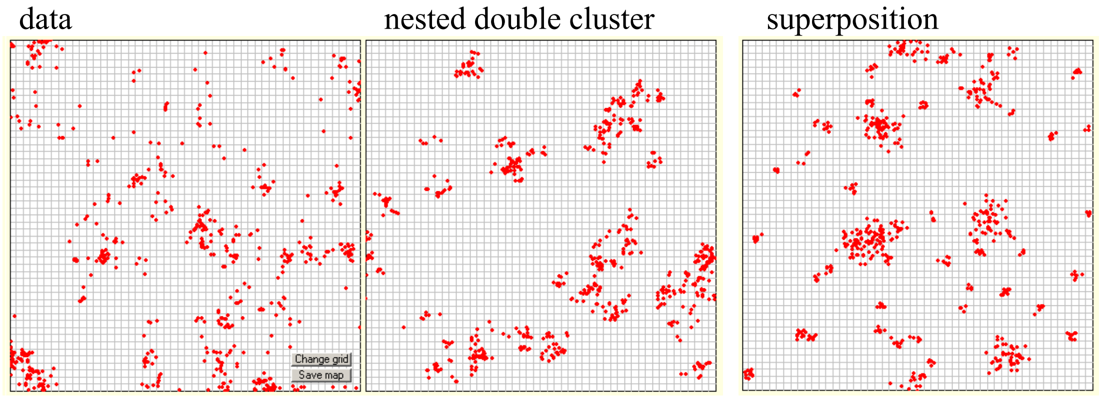

Simulation

□ neg. Binom, k= 0.5 9999
□ # p large cluster 313 ☑ 2 Thomas

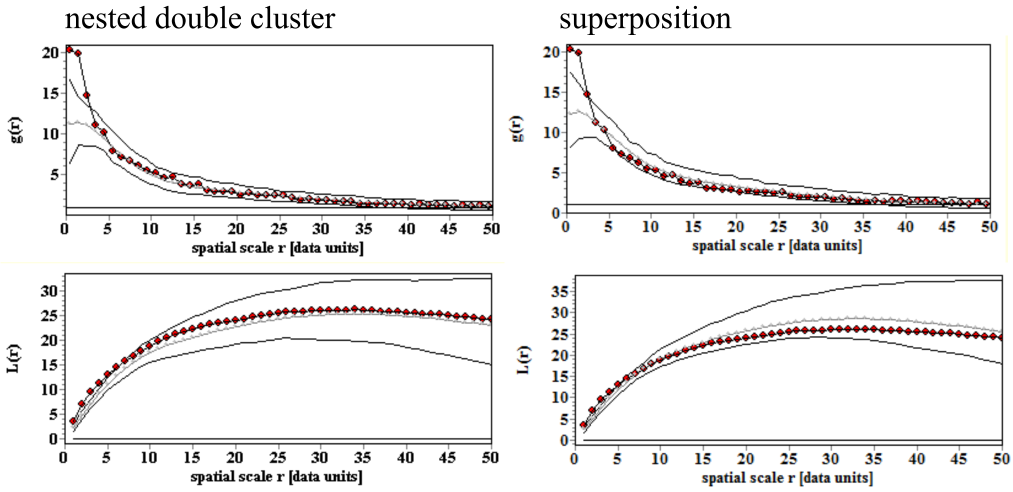
and finally click the small "**ok"** button at the window "Fitted parameters".

3. After pressing the "**Calculate Index**" button *Programita* simulates the superposition point process. Comparison between the simulated nested double-cluster process and the superposition process show clear differences.

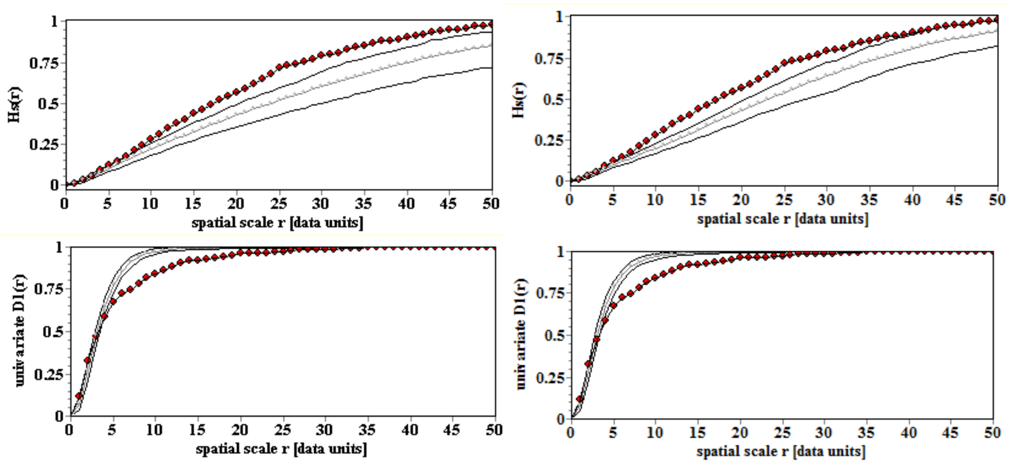
The superposition process shows many small clusters (49 vs. 198 for the double cluster process) and few large clusters (7 vs. 31 for the double cluster process):



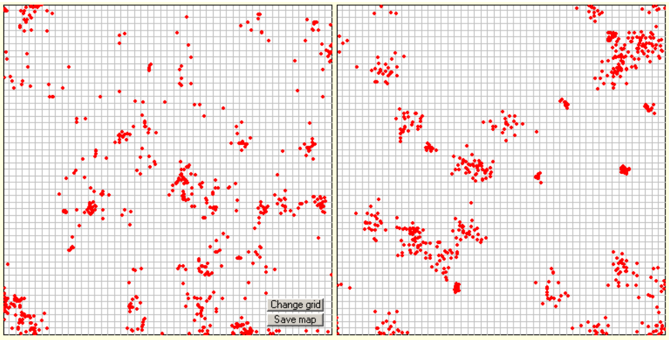
As expected, the fit of the pair correlation function and the *L*-function of the superposition and the nested process are similar:



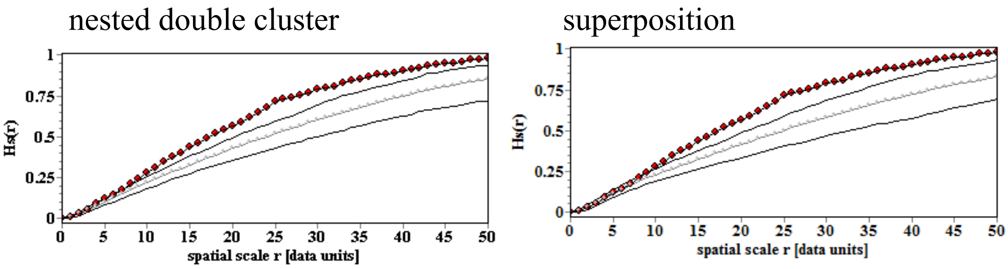
Interestingly, the fit for the spherical contact distribution and the nearest neighbor distribution function are also similar:



4. Using instead a superposition process with a large-scale cluster process with 500 points (and 126 points being part of the small-scale cluster process) we have 20 large clusters but 8 small clusters:

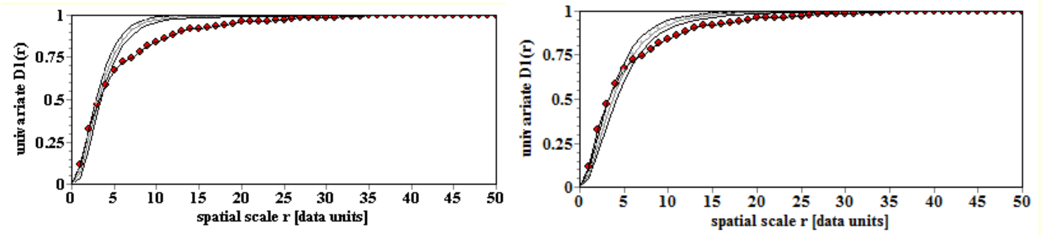


5. The spherical contact distributions of the nested double cluster and the superposition process are similar:


nested double cluster

superposition

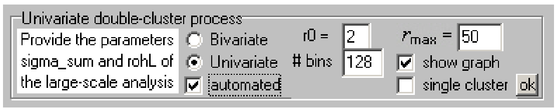
but the nearest neighbor distribution function changes:



Examples TwoThomas313.res (superposition of two Thomas processes)

In this example we use a pattern generated with the fitted superposition process in example Fig4_15super.res where both component processes had the same number of points (= 313). We fit this pattern to the superposition process of two simple Thomas processes.

1. Execute *Programita*.
2. Highlight data file TwoThomas313.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
5. Press button "**Calculate Index**"
6. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
7. Select "**Cluster process**" in the window "Select a null model".
8. A window "Fit of cluster process to data" opens. Select in the section "Null models" at the bottom "**Univar. cluster**". Next enable "Univariate" and "automated":



Univariate double-cluster process

Provide the parameters sigma_sum and rohL of the large-scale analysis

○ Bivariate

● Univariate

☑ automated

r0 = 2

bins 128

r_{max} = 50

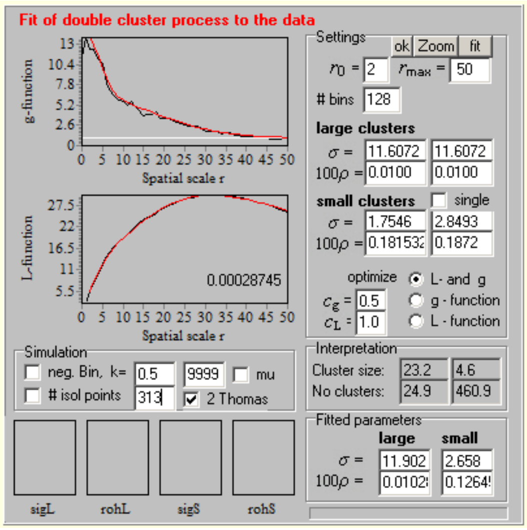
☑ show graph

□ single cluster

ok

Select also the distance interval (r_0, r_{max}) used for fitting. You can also change the number of bins used for each parameter in the σ-ρ parameter space. Note that the time requirement for the fit scales with (# bins)^2. Finally click the small "**ok**" button to start the automatic fitting procedure.

9. Once Programita has finished this task, this window is shown:



Fit of double cluster process to the data

Settings

ok Zoom fit

r_0 = 2 r_{max} = 50

bins 128

large clusters

σ = 11.6072 11.6072

100ρ = 0.0100 0.0100

small clusters □ single

σ = 1.7546 2.8493

100ρ = 0.181532 0.1872

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Simulation

□ neg. Bin, k= 0.5 9999 □ mu

□ # isol points 313 ☑ 2 Thomas

Interpretation

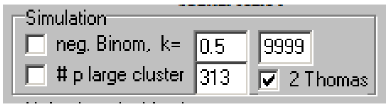
Cluster size: 23.2 4.6

No clusters: 24.9 460.9

Fitted parameters

	large	small
σ =	11.902	2.658
100ρ =	0.0102!	0.1264!

10. Select the checkbox "2 Thomas" and insert true number of points of large scale clustering (313):



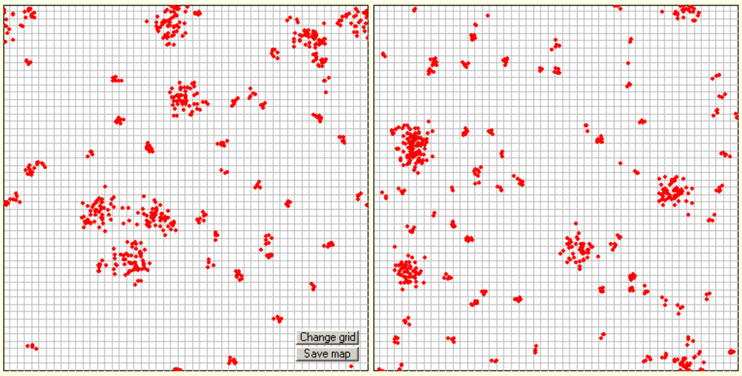
Simulation

□ neg. Binom, k= 0.5 9999

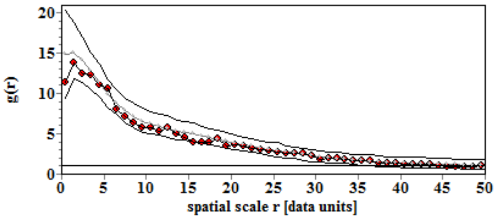
□ # p large cluster 313 ☑ 2 Thomas

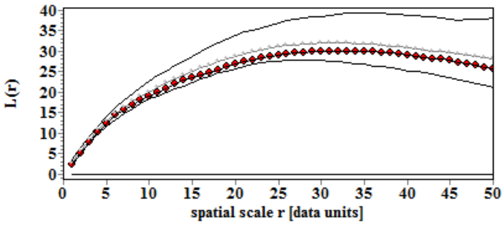
11. Finally, click the small **"ok"** button and then "**Calculate Index**" to simulate the fitted point process.

12. As expected, the simulated and the observed pattern are very similar, we have 6 large clusters and 77 small clusters:

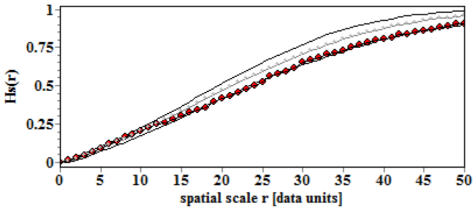


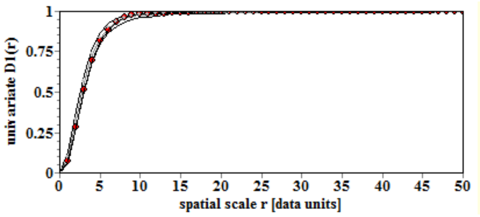
As expected, the pair correlation function and the L function are well fitted:





The spherical contact distribution is slightly overestimated but the distribution function of the nearest neighbor distances is well fitted:





The parameters of the original point process are reasonably recovered: σ_S = 2.66 (vs. 3.64), σ_L = 11.9 (vs. 13.3), $A\rho_S$ = 461 (vs. 199), and $A\rho_L$ = 24.9 (vs. 31.5). Because of the stochasticity of the point process and the low number of clusters we cannot expect a better fit between parameters used to simulate the point process and the fitted parameters of a realization of the point process.

As exercise you can simulate the superposition process with different numbers of points of the large scale clustering. With 500 points (*TwoThomas313_500.res*) the spherical contact distribution is now underestimated and the distribution function of the nearest neighbor distances is also underestimated.

With 250 points (*TwoThomas313_250.res*) the spherical contact distribution is now substantially overestimated but the distribution function of the nearest neighbor distances is well fitted.

3.3.9 Series of analysis with cluster point processes

A common task is to assess the clustering of several (plant) species in a community. In this case you need to fit a Thomas cluster point processes to the pattern of each species. To avoid the tedious task of conducting the fit for each species individually, *Programita* allows you to do this in an automated way. To this end you can select the files to be analyzed from a list and fit the three basic Thomas processes

- simple univariate Thomas process with one critical scale of clustering (with a Poisson distribution governing the distribution of the points over the clusters)
- the generalized Thomas process with two nested scale of clustering (with a Poisson distribution governing the distribution of the points over the clusters)
- the superposition of two simple Thomas processes (with Poisson distributions governing the distribution of the points over the clusters)

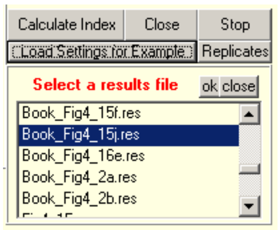
to the data based on the second-order summary functions $g(r)$ and $L(r)$. However, as show in Wiegand and Moloney (2014: sections 4.1.4.2, 4.1.4.5, and Figs. 4.11 and 4.15), although the second-order summary functions remain unchanged, different the distribution of the points of the clusters can produce patterns with largely different structure captured by the spherical contact distribution $H_s(r)$ and the nearest neighbor distribution function $D(r)$. Failure to account for such differences can severely limit the inference made by fitting cluster processes to the data. Therefore, you can additionally fit more detailed features of the Thomas processes not determined by the second-order summary functions:

- change the distribution of points over the clusters from a Poisson distribution (random assignment to a cluster) to a negative Binominal distribution (clustered assignment)
- assume an independent superposition with a CSR pattern

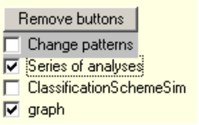
This results in a wide range of point processes that allows for describing the detailed features of clustering for a wide range of clustered data. *Programita* conducts many individual analyses, outputs results files for each analysis, and one summary file that provides an overview over all analyses.

Step-by-step example for series of analyses using the generalized Thomas process with two scales of clustering

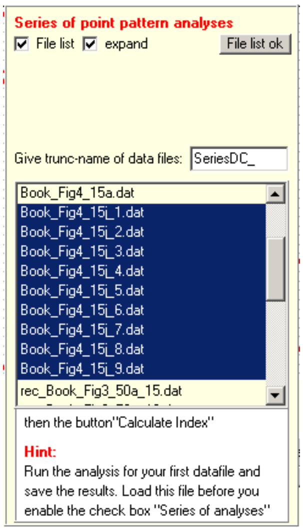
The example uses point patterns generated by example *Book_Fig4_15j.res* which generates patterns with two nested scales of clustering ($\sigma_S = 3.7$, $\sigma_L = 14.2$) and clustered assignment of points to the clusters ($k_S = 0.1$, $k_L = 0.1$).



1. The first step is to conduct the analysis with one of the data files. Use for example the data file from analysis “Book_Fig4_15j.res” that was used to generate the patterns. Read the settings with “**Load settings for Example**” and repeat the analysis with “**Calculate Index**”
2. Once this is done, select the check box “**Series of analysis**”. There are nine data file with names “Book_Fig4_15j_1.dat”, , ..., “Book_Fig4_15j_9.dat” to be analyzed in the same way as the original data file “book_Fig4_15a.dat” used in the example.



3. A window opens where you need to provide the specifications of your series of analysis. First click “File list” and “expand” to select the data files to be analyzed. Once all data files are highlighted click the small button “File list ok” and select a name for the analysis (e.g., SeriesDC_”). The output results files will be named based on this name.



Series of point pattern analyses

☑ File list ☑ expand

Give trunc-name of data files: SeriesDC_

Book_Fig4_15a.dat
Book_Fig4_15j_1.dat
Book_Fig4_15j_2.dat
Book_Fig4_15j_3.dat
Book_Fig4_15j_4.dat
Book_Fig4_15j_5.dat
Book_Fig4_15j_6.dat
Book_Fig4_15j_7.dat
Book_Fig4_15j_8.dat
Book_Fig4_15j_9.dat
rec_Book_Fig3_50a_15.dat

then the button"Calculate Index"

Hint:
Run the analysis for your first datafile and save the results. Load this file before you enable the check box "Series of analyses"

4. Disable “save bi_confidence” because you run an univariate analysis

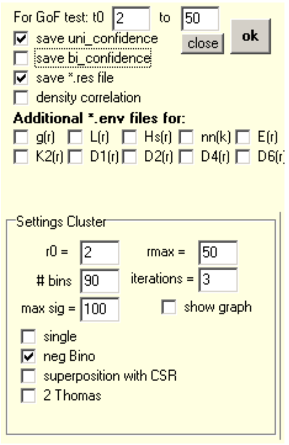
5. Provide distance interval (r_0, $r_{\max}$) for the GoF tests. In the example it is $r_0 = 2$ and $r_{\max} = 50$.

6. Go to the box “**Settings cluster**” to select details of the cluster analysis. Provide distance interval (r_0, $r_{\max}$) for the fitting. In the example it is $r_0 = 2$ and $r_{\max} = 50$ (the same as for the GoF tests).

7. Select “**# bins**”, the number of intervals the interval of the parameters is derived into. *Programita* tests the entire grid of the (σ, ρ) parameter space (i.e., [# bins]^2 values are tested). The large the value the more precise is the estimate of the parameters but the longer the time requirement which increases with [# bins]^2. Select for example 90 bins.

8. Select “**max sig**” which gives the maximal value for parameter σ (100 in the example).

9. Select “**iterations**”. This value determines the number of “zooms” into (or out of) the parameter space to get fines parameter estimates. A value of 3 iterations produces good results, if you select more the procedure is slower.



For GoF test: t0 2 to 50

☑ save uni_confidence
□ save bi_confidence
☑ save *.res file
□ density correlation

Additional *.env files for:

□ g(r) □ L(r) □ Hs(r) □ nn(k) □ E(r)
□ K2(r) □ D1(r) □ D2(r) □ D4(r) □ D6(r

Settings Cluster

r0 = 2 rmax = 50
# bins 90 iterations = 3
max sig = 100 □ show graph

□ single
☑ neg Bino
□ superposition with CSR
□ 2 Thomas

10. Select “**single**” only if you want to fit a single cluster process to the data.

11. If you select “**neg Bino**”, *Programita* will first fit the standard Thomas process with two scales of clustering to the data (or with one scale of clustering if “single was selected”) and then simulate the patterns for ten different values of the parameters k_L and k_S that govern how the points are distributed over the large and small clusters, respectively. We use here values of $k_L = k_S = 1000, 5, 1, 0.3, 0.1, 0.05, 0.03, 0.015, 0.0083, 0.0045, 0.0025, 0.0014$. Other options (e.g., $k_L = 1000$, and k_S variable) can be implemented if needed. This option works also together with the option “**single**”.

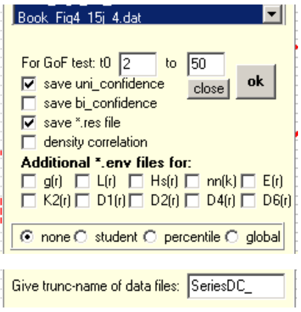
12. If you select “**superposition with CSR**” *Programita* will first fit the standard Thomas process with two scales of clustering to the data (or with one scale of clustering if “single was selected”) and then simulate the patterns for ten proportions of points being part of a CSR component pattern. The maximal percentage of CSR points can be selected with “**max. percent CSR points**”.

13. If you select “**2 Thomas**” *Programita* will first fit the standard Thomas process with two scales of clustering to the data (or with one scale of clustering if “single was selected”) and then simulate the patterns of the corresponding superposition of two Thomas processes with one scale of clustering for ten different values of proportion of points being part of the large scale Thomas process.

14. If all settings are correct, click the large “ok” button and then “Calculate index”.

Programita outputs for each analysis a number of files:

1. if "**save *.res file**" was enabled Programita saves *.res file that contains all settings of this particular analysis. In the example, because "**neg Bino**" was enabled, *Programita* saves ten numbered results files for each data file because ten different values for k_L and k_S were tested. They are named for the nth data file "SeriesDC_n_1.res", "SeriesDC_n_10.res" (based on the name you select in "**Give trunk-name of data file**").



Book Fig4 15j 4.dat ▾

For GoF test: t0 2 to 50

☑ save uni_confidence

□ save bi_confidence

☑ save *.res file

□ density correlation

Additional *.env files for:

□ g(r) □ L(r) □ Hs(r) □ nn(k) □ E(r)

□ K2(r) □ D1(r) □ D2(r) □ D4(r) □ D6(r)

● none ○ student ○ percentile ○ global

Give trunc-name of data files: SeriesDC_

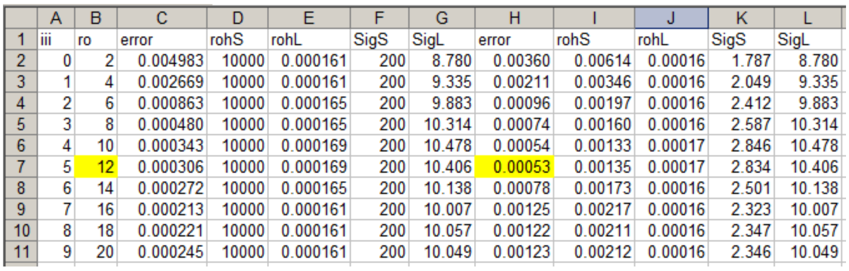
2. if "**save uni_confidence**" was selected, *Programita* saves for each data file (numbered consecutively) the files "Uni_confidence1.env", …"Uni_confidence10.env".

3. If any of the "**Additional *.env files for**" was selected, *Programita* outputs the *.env files for each the selected summary functions for each analysis. They are named for the nth data file:

   - g(r)_SeriesDC_n_1.env,.., g(r)_SeriesDC_n_10.env
   - L(r)_SeriesDC_n_1.env,.., L(r)_SeriesDC_n_10.env
   - D1(r)_SeriesDC_n_1.env,…., D1(r)_SeriesDC_n_10.env
   - Hs(r)_SeriesDC_n_1.env,…., Hs(r)_SeriesDC_n_10.env

   …

4. To document the fit, *Programita* saves for the nth data file a file fitDC_dataname_n.txt that shows the best fitting parameters for the different values of the auxiliary parameter r_{0L} that gives the distance r_{0L} where the small-scale clustering component just disappears. In the example we find that a value of r_{0L} = 12 produces the smallest error and was therefore selected:

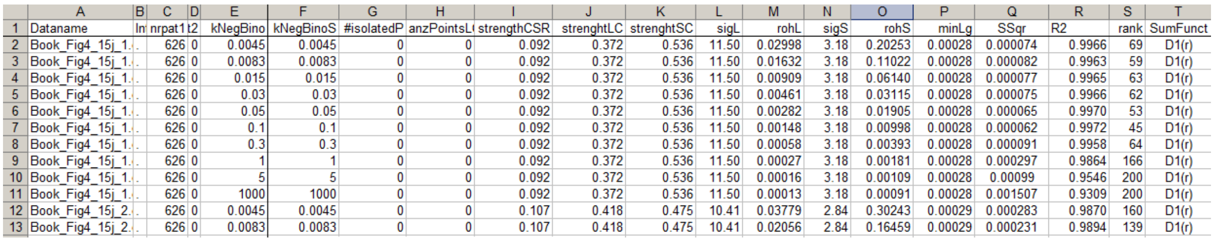


	A	B	C	D	E	F	G	H	I	J	K	L
1	iii	ro	error	rohS	rohL	SigS	SigL	error	rohS	rohL	SigS	SigL
2	0	2	0.004983	10000	0.000161	200	8.780	0.00360	0.00614	0.00016	1.787	8.780
3	1	4	0.002669	10000	0.000161	200	9.335	0.00211	0.00346	0.00016	2.049	9.335
4	2	6	0.000863	10000	0.000165	200	9.883	0.00096	0.00197	0.00016	2.412	9.883
5	3	8	0.000480	10000	0.000165	200	10.314	0.00074	0.00160	0.00016	2.587	10.314
6	4	10	0.000343	10000	0.000169	200	10.478	0.00054	0.00133	0.00017	2.846	10.478
7	5	12	0.000306	10000	0.000169	200	10.406	0.00053	0.00135	0.00017	2.834	10.406
8	6	14	0.000272	10000	0.000165	200	10.138	0.00078	0.00173	0.00016	2.501	10.138
9	7	16	0.000213	10000	0.000161	200	10.007	0.00125	0.00217	0.00016	2.323	10.007
10	8	18	0.000221	10000	0.000161	200	10.057	0.00122	0.00211	0.00016	2.347	10.057
11	9	20	0.000245	10000	0.000161	200	10.049	0.00123	0.00212	0.00016	2.346	10.049

5. Finally, *Programita* outputs one large table that summarizes the results of the series of analyses named in the example "Summary_FitDCseriesDC_.txt". …

The results table

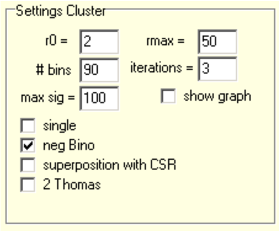
Each line of the results table corresponds to the results of one analysis for one summary function out of the four summary functions $g(r)$, $L(r)$, $D(r)$ and $H_s(r)$. In the example we have nine data files, ten values of k_S, and four summary functions which results in 9×10×4 = 360 lines. The lines are originally ordered by data file (Dataname), k_S (kNegBino) and summary function (SumFunct) but it is better to order the table by SumFunct, Dataname, kNegBino.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Dataname	In	nrpat1	t2	kNegBino	kNegBinoS	#isolatedP	anzPointsL	strengthCSR	strenghtLC	strenghtSC	sigL	rohL	sigS	rohS	minLg	SSqr	R2	rank	SumFunct
2	Book_Fig4_15j_1.	.	626	0	0.0045	0.0045	0	0	0.092	0.372	0.536	11.50	0.02998	3.18	0.20253	0.00028	0.000074	0.9966	69	D1(r)
3	Book_Fig4_15j_1.	.	626	0	0.0083	0.0083	0	0	0.092	0.372	0.536	11.50	0.01632	3.18	0.11022	0.00028	0.000082	0.9963	59	D1(r)
4	Book_Fig4_15j_1.	.	626	0	0.015	0.015	0	0	0.092	0.372	0.536	11.50	0.00909	3.18	0.06140	0.00028	0.000077	0.9965	63	D1(r)
5	Book_Fig4_15j_1.	.	626	0	0.03	0.03	0	0	0.092	0.372	0.536	11.50	0.00461	3.18	0.03115	0.00028	0.000075	0.9966	62	D1(r)
6	Book_Fig4_15j_1.	.	626	0	0.05	0.05	0	0	0.092	0.372	0.536	11.50	0.00282	3.18	0.01905	0.00028	0.000065	0.9970	53	D1(r)
7	Book_Fig4_15j_1.	.	626	0	0.1	0.1	0	0	0.092	0.372	0.536	11.50	0.00148	3.18	0.00998	0.00028	0.000062	0.9972	45	D1(r)
8	Book_Fig4_15j_1.	.	626	0	0.3	0.3	0	0	0.092	0.372	0.536	11.50	0.00058	3.18	0.00393	0.00028	0.000091	0.9958	64	D1(r)
9	Book_Fig4_15j_1.	.	626	0	1	1	0	0	0.092	0.372	0.536	11.50	0.00027	3.18	0.00181	0.00028	0.000297	0.9864	166	D1(r)
10	Book_Fig4_15j_1.	.	626	0	5	5	0	0	0.092	0.372	0.536	11.50	0.00016	3.18	0.00109	0.00028	0.00099	0.9546	200	D1(r)
11	Book_Fig4_15j_1.	.	626	0	1000	1000	0	0	0.092	0.372	0.536	11.50	0.00013	3.18	0.00091	0.00028	0.001507	0.9309	200	D1(r)
12	Book_Fig4_15j_2.	.	626	0	0.0045	0.0045	0	0	0.107	0.418	0.475	10.41	0.03779	2.84	0.30243	0.00029	0.000283	0.9870	160	D1(r)
13	Book_Fig4_15j_2.	.	626	0	0.0083	0.0083	0	0	0.107	0.418	0.475	10.41	0.02056	2.84	0.16459	0.00029	0.000231	0.9894	139	D1(r)

The results file contains the following columns:

1. **Dataname**: the name of the data file containing the coordinates of the pattern to be analyzed
2. **Intname**: the name of the intensity file associated with the pattern. Only used if inhomogeneous second-order summary functions are used.
3. **nrpat1**, **nrpat2**: number of points of pattern 1 and 2. For univariate analysis we have nrpat2 = 0.
4. **kNegBino**, **kNegBinoS**: the values of k_S and k_L of the negative Binominal distribution. A value of 1000 corresponds to the standard case of the Poisson distribution. The columns are only relevant if "**neg Bino**" was enabled.



Settings Cluster

r0 = 2 rmax = 50

bins 90 iterations = 3

max sig = 100 □ show graph

□ single

☑ neg Bino

□ superposition with CSR

□ 2 Thomas

5. **#isolatedP**: the number of points belonging to the CSR component pattern. The columns is only relevant "**superposition with CSR**" was enabled.
6. **anzPointsLC**: the number of points belonging to the large-scale Thomas process. Only relevant if "**2 Thomas**" was enabled.
7. **strengthCSR**, **strenghtLC**, **strenghtSC:** measures of the relative contribution of CSR, large scale clustering and small-scale clustering to the value of the pair correlation function $g(r_0)$. The three values add up to one and are independent of k_S and $k_{L.}$

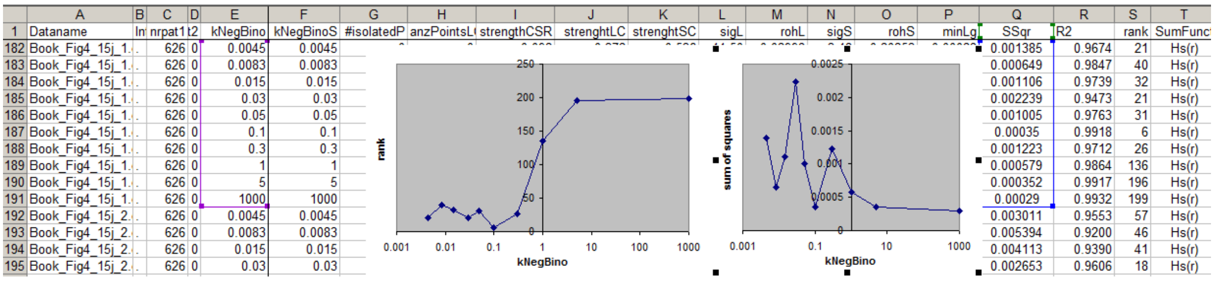
   We define strengthCSR = $1/g(r_0)$, strenghtLC = stLC/$g(r_0)$, and strenghtSC = stSC/$g(r_0)$

   where $g(r_0) = 1 + \underbrace{\frac{f_{k_L}}{\rho_L} \frac{e^{-r_0^2/4\sigma_L^2}}{4\pi\sigma_L^2}}_{\text{stLC}} + \underbrace{\frac{f_{k_S}}{\rho_S} \frac{e^{-r_0^2/4(\sigma_L^2+\sigma_S^2)}}{4\pi(\sigma_L^2+\sigma_S^2)}}_{\text{stSC}}$ and $f_{k_L} = \frac{(k_L - 1)}{k_L}, f_{k_S} = \frac{(k_S - 1)}{k_S}$.

8. **sigL**, **rohL**, **sigS**, **rohS**: the fitted parameters σ_L, σ_S, ρ_L, ρ_S.
9. **minLg**: The value of the discrepancy *error_Lg* between the observed and fitted second-order summary functions $L(r)$ and $g(r)$. Parameter fitting minimizes *minLG*.
10. **SSqr**: The sum of squares between the observed summary function and the expected summary function of the fitted point process over the distance interval (r_0, $r_{\max}$) which was (2, 50) in the example.
11. **R2**: The proportion of variation explained by the model over the distance interval (r_0, $r_{\max}$), calculated as 1-SSE/SST, where SSE=$\Sigma_r[F_{ob}(r)$ - $F_{pr}(r)]^2$ and SST= $\Sigma_r[F_{ob}(r)$ - mean$(F_{ob})]^2$, $F_{pr}(r)$ is the predicted summary function and $F_{ob}(r)$ is the observed one. Note that $R^2 < 0$ if the intercept only model (i.e. mean(F_{ob})) fits better than the model.

12. **rank**: the rank of the Loosmore and Ford (2006) GoF test. For the 199 simulations of the point process model.
13. **SumFunct**: the summary function, being $g(r)$, $L(r)$, $D(r)$, or $H_s(r)$.

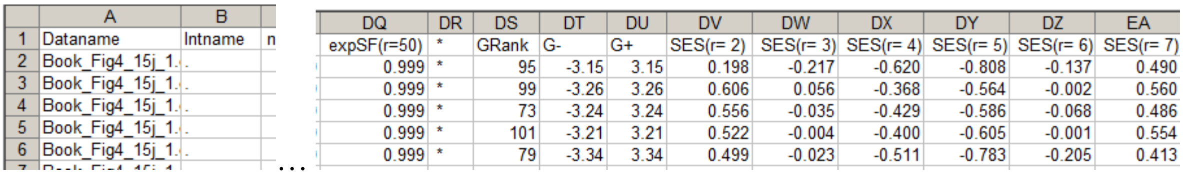
In the example, we varied the parameter k_S and k_L of the negative Binominal distribution to obtain non-random (i.e., non-Poisson) variability in the number of points per cluster. Because the values of second-order summary functions are not affected by the way how the points are distributed over the clusters, we need to use the nearest neighbor summary functions $D(r)$ and $H_s(r)$ to determine the values of k_S and k_L that produce the best fit. The results of the GoF test for the $H_s(r)$ shows that the value $k_S = k_L = 0.1$ produces the best fit, and the values $k_S = k_L = 0.1$ produce also small sum of squares. Thus, we were able to recover the values of k_S and k_L used for simulation of the pattern.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Dataname	In	nrpat1	t2	kNegBino	kNegBinoS	#isolatedP	anzPointsL	strengthCSR	strenghtLC	strenghtSC	sigL	rohL	sigS	rohS	minLg	SSqr	R2	rank	SumFunct
182	Book_Fig4_15j_1.	.	626	0	0.0045	0.0045											0.001385	0.9674	21	Hs(r)
183	Book_Fig4_15j_1.	.	626	0	0.0083	0.0083											0.000649	0.9847	40	Hs(r)
184	Book_Fig4_15j_1.	.	626	0	0.015	0.015											0.001106	0.9739	32	Hs(r)
185	Book_Fig4_15j_1.	.	626	0	0.03	0.03											0.002239	0.9473	21	Hs(r)
186	Book_Fig4_15j_1.	.	626	0	0.05	0.05											0.001005	0.9763	31	Hs(r)
187	Book_Fig4_15j_1.	.	626	0	0.1	0.1											0.00035	0.9918	6	Hs(r)
188	Book_Fig4_15j_1.	.	626	0	0.3	0.3											0.001223	0.9712	26	Hs(r)
189	Book_Fig4_15j_1.	.	626	0	1	1											0.000579	0.9864	136	Hs(r)
190	Book_Fig4_15j_1.	.	626	0	5	5											0.000352	0.9917	196	Hs(r)
191	Book_Fig4_15j_1.	.	626	0	1000	1000											0.00029	0.9932	199	Hs(r)
192	Book_Fig4_15j_2.	.	626	0	0.0045	0.0045											0.003011	0.9553	57	Hs(r)
193	Book_Fig4_15j_2.	.	626	0	0.0083	0.0083											0.005394	0.9200	46	Hs(r)
194	Book_Fig4_15j_2.	.	626	0	0.015	0.015											0.004113	0.9390	41	Hs(r)
195	Book_Fig4_15j_2.	.	626	0	0.03	0.03											0.002653	0.9606	18	Hs(r)

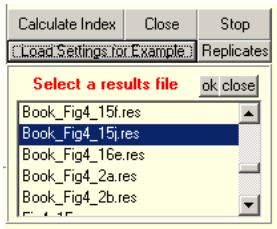
For a more detailed assessment of the fit, the results file contains the following additional columns:

14. **obsSF(r= 0), …, obsSF(r= rmax)**: the values of the observed summary function
15. **expSF(r= 0), …, expSF(r= rmax)**: the values of the summary function expected under the point process model, being the mean of that of the simulated patterns.
16. **GRank**: the rank of the global envelope test conducted for the given summary function over the distance interval (r_0, .., $r_{\max}$).
17. **G-, G+**: the lower and upper global envelope of the global test of the student-transformed summary function.
18. **SES($\mathbf{r_0}$), …, SES($\mathbf{r_{0max}}$)**: the standardized effect sizes of the summary functions for the distance interval (r_0, .., $r_{\max}$).



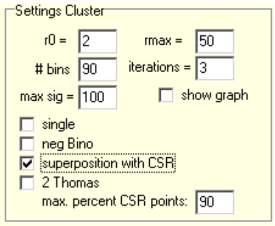
	A	B	…	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA
1	Dataname	Intname	n	expSF(r=50)	*	GRank	G-	G+	SES(r= 2)	SES(r= 3)	SES(r= 4)	SES(r= 5)	SES(r= 6)	SES(r= 7)
2	Book_Fig4_15j_1.	.		0.999	*	95	-3.15	3.15	0.198	-0.217	-0.620	-0.808	-0.137	0.490
3	Book_Fig4_15j_1.	.		0.999	*	99	-3.26	3.26	0.606	0.056	-0.368	-0.564	-0.002	0.560
4	Book_Fig4_15j_1.	.		0.999	*	73	-3.24	3.24	0.556	-0.035	-0.429	-0.586	-0.068	0.486
5	Book_Fig4_15j_1.	.		0.999	*	101	-3.21	3.21	0.522	-0.004	-0.400	-0.605	-0.001	0.554
6	Book_Fig4_15j_1.	.		0.999	*	79	-3.34	3.34	0.499	-0.023	-0.511	-0.783	-0.205	0.413

Step-by-step example for series of analyses using the superposition of a Thomas process with two scales of clustering with a random pattern



The example repeats the above example but uses the superposition of a Thomas process with two scales of clustering with a random pattern instead. The example is based on the point patterns generated by example *Book_Fig4_15j.res* which generates patterns with two nested scales of clustering ($\sigma_S = 3.7$, $\sigma_L = 14.2$) and clustered assignment of points to the clusters ($k_S = 0.1$, $k_L = 0.1$).

All steps 1 - 14 of the above example are identical, the only difference is that you do not select "**neg Bino**" (step 11) but "**superposition with CSR**" (step 12):



Settings Cluster
r0 = 2 rmax = 50
# bins 90 iterations = 3
max sig = 100 □ show graph
□ single
□ neg Bino
☑ superposition with CSR
□ 2 Thomas
max. percent CSR points: 90

12. If you select "**superposition with CSR**" *Programita* will first fit the standard Thomas process with two scales of clustering to the data (or with one scale of clustering if "single was selected") and then simulate the patterns for ten proportions of points being part of a CSR component pattern. The maximal percentage of CSR points can be selected with "**max. percent CSR points**".
13. The example patterns Book_Fig4_15j_n.dat have 626 points, thus with a maximal percentage of 90% CSR points you have in the extreme case 63 points of the Thomas process and 563 random points.

The results table is also identical, only the estimation of the columns **strengthCSR**, **strenghtLC**, **strenghtSC** differ because of the slightly analytical expressions of the pair correlation functions of two point processes.

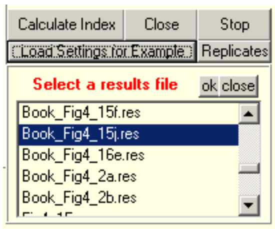
7. **strengthCSR**, **strenghtLC**, **strenghtSC:** measures of the relative contribution of CSR, large scale clustering and small-scale clustering to the value of the pair correlation function $g(r_0)$. The three values add up to one and are independent of the parameter p_C of the superposition process, which is the proportion of points of the pattern belonging to the double-cluster Thomas process:

   We define strengthCSR = $1/g(r_0)$, strenghtLC = stLC/$g(r_0)$, and strenghtSC = stSC/$g(r_0)$

   where $$g(r_0) = 1 + \underbrace{\frac{p_C^2}{\rho_L}\,\frac{e^{-r_0^2/4\sigma_L^2}}{4\pi\sigma_L^2}}_{\text{stLC}} + \underbrace{\frac{p_C^2}{\rho_S}\,\frac{e^{-r_0^2/4(\sigma_L^2+\sigma_S^2)}}{4\pi(\sigma_L^2+\sigma_S^2)}}_{\text{stSC}} \quad .$$

Step-by-step example for series of analyses using the superposition of two single cluster Thomas processes

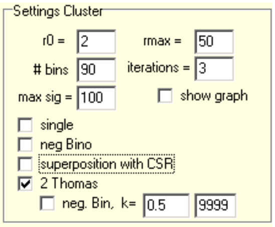
The example repeats the above example but uses the superposition of a Thomas process with two scales of clustering with a random pattern instead. The example is based on the point patterns generated by example *Book_Fig4_15j.res* which generates patterns with two nested scales of clustering ($\sigma_S = 3.7$, $\sigma_L = 14.2$) and clustered assignment of points to the clusters ($k_S = 0.1$, $k_L = 0.1$).



Book_Fig4_15f.res
Book_Fig4_15j.res
Book_Fig4_16e.res
Book_Fig4_2a.res
Book_Fig4_2b.res

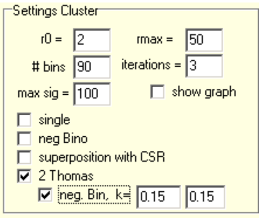
All steps 1 - 14 of the above example are identical, the only difference is that you do not select "**neg Bino**" (step 11) but "**2 Thomas**" (step 13):

13. If you select "**2 Thomas**" *Programita* will first fit the standard Thomas process with two scales of clustering to the data (or with one scale of clustering if "single was selected") and then simulate the patterns of the corresponding superposition of two Thomas processes with one scale of clustering for ten different values of proportion of points being part of the large scale Thomas process. The range of this proportion is defined in a way that the patterns of two extreme cases comprise two or more large or small clusters. To use a negative Binominal distribution to distribute the points over the clusters click "**neg. Bin**".



r0 = 2 rmax = 50
# bins 90 iterations = 3
max sig = 100 □ show graph
□ single
□ neg Bino
□ superposition with CSR
☑ 2 Thomas
□ neg. Bin, k= 0.5 9999

14. If you want to use generalized Thomas processes with one scale of clustering where the number of points per cluster follows a negative Bionomical distribution select "**neg. Bin**" and provide the k-values of the two distributions.



r0 = 2 rmax = 50
# bins 90 iterations = 3
max sig = 100 □ show graph
□ single
□ neg Bino
□ superposition with CSR
☑ 2 Thomas
☑ neg. Bin, k= 0.15 0.15

The results table is also identical, only the estimation of the columns **strengthCSR**, **strenghtLC**, **strenghtSC** differ because of the slightly analytical expressions of the pair correlation functions of two point processes.

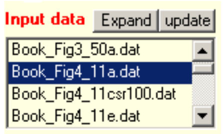
7. **strengthCSR**, **strenghtLC**, **strenghtSC:** measures of the relative contribution of CSR, large scale clustering and small-scale clustering to the value of the pair correlation function $g(r_0)$. The three values add up to one and are independent of the parameter p_C of the superposition process, which is the proportion of points of the pattern belonging to the double-cluster Thomas process:

   We define strengthCSR = $1/g(r_0)$, strenghtLC = stLC/$g(r_0)$, and strenghtSC = stSC/$g(r_0)$

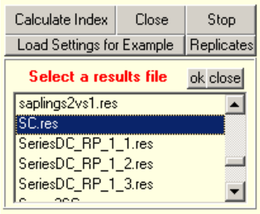
   where $g(r_0) = 1 + \underbrace{\frac{(1-p_L)^2}{\rho_S^s}\frac{e^{-r_0^2/4\sigma_S^2}}{4\pi\sigma_S^2}}_{\text{stSC}} + \underbrace{\frac{p_L^2}{\rho_L^s}\frac{e^{-r_0^2/4(\sigma_L^s)^2}}{4\pi(\sigma_L^s)^2}}_{\text{stLC}}$ .and $\sigma_L{}^s$, $\rho_L{}^s$, $\rho_L{}^s$ are the parameters of the superposition process that need to be transformed with respect to the parameters fitted with the procedure for the nested double cluster Thomas process.

Step-by-step example for series of analyses using a single cluster Thomas processes

The example repeats the above example but uses the Thomas process with two one of clustering. The example is based on the point pattern *Book_Fig4_12a.dat* which was generated with one nested scales of clustering ($\sigma_L = 6$, $\rho_L = 0.02$) and random assignment of points to the clusters ($k_S = 9999$, $k_L = 9999$). The fitted point process had very similar parameters ($\sigma_L = 5.91$, $\rho_L = 0.0221$) and the patterns SC_1.dat, ..., SC_9.dat are generated with the latter parameter set.

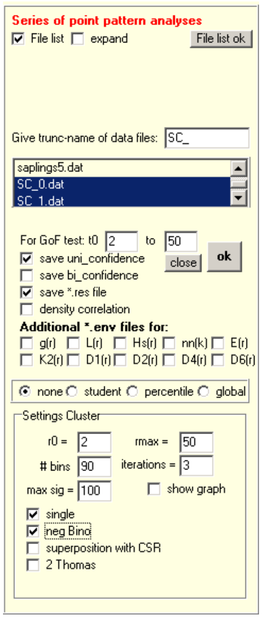


Book_Fig3_50a.dat
Book_Fig4_11a.dat
Book_Fig4_11csr100.dat
Book_Fig4_11e.dat



saplings2vs1.res
SC.res
SeriesDC_RP_1_1.res
SeriesDC_RP_1_2.res
SeriesDC_RP_1_3.res

All steps 1 - 14 of the above example are identical, use the settings file SC.rep to read the parameters of this point process. The only difference is that you do select additionally "**single**" and then "**neg Bino**" to test the generalized Thomas process with one scale of clustering and a negative Binominal distribution governing the distribution of the number of points over the clusters.



☑ File list □ expand

Give trunc-name of data files: SC_

saplings5.dat
SC_0.dat
SC_1.dat

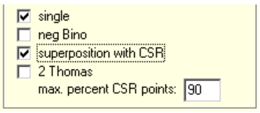
For GoF test: t0 2 to 50
☑ save uni_confidence
□ save bi_confidence
☑ save *.res file
□ density correlation

□ g(r) □ L(r) □ Hs(r) □ nn(k) □ E(r)
□ K2(r) □ D1(r) □ D2(r) □ D4(r) □ D6(r)

● none ○ student ○ percentile ○ global

r0 = 2 rmax = 50
# bins 90 iterations = 3
max sig = 100 □ show graph
☑ single
☑ neg Bino
□ superposition with CSR
□ 2 Thomas

10. If you select "**single**" and "**neg Bino**" *Programita* will first fit the standard Thomas process with one scale of clustering to the data and then simulate the patterns of the corresponding generalized Thomas processes with one scale of clustering for ten different values of the parameters k_L that govern how the points are distributed over the clusters. We use here values of $k_L = k_S = 1000$, 5, 1, 0.3, 0.1, 0.05, 0.03, 0.015, 0.0083, 0.0045, 0.0025, 0.0014 that cover the entire range of parameter values.
11. If you select "**single**" and "**superposition with CSR**" *Programita* will first fit the standard Thomas process with one scale of clustering to the and then simulate the patterns for ten proportions of points being part of a CSR component pattern. The maximal percentage of CSR points can be selected with "**max. percent CSR points**".



☑ single
□ neg Bino
☑ superposition with CSR
□ 2 Thomas
max. percent CSR points: 90

12. The example pattern *Book_Fig4_12a.dat* has 626 points, thus with a maximal percentage of 90% CSR points you have in the extreme case 63 points of the Thomas process and 563 random points.

3.4 Inhomogeneous *g*- and *K* functions

3.4.1 Estimators of inhomogeneous *g*- and *K* functions

Inhomogeneous second-order statistics can be used in *Programita* for two different purposes. First, they offer a natural estimator of the second-order summary functions **for irregularly shaped observation windows**. In this case the *.int file that defines the intensity function $\lambda(\mathbf{x})$ contains only the values “1” (inside the observation window) and “0” (outside the observation window). Second, the second-order statistics can be used to **remove the effect of environmental heterogeneity** represented by an intensity function $\lambda(\mathbf{x})$. The inhomogeneous *g*- and *K* functions show then the residual clustering or overdispersion, conditionally on $\lambda(\mathbf{x})$.

You can use in *Programita* two different estimators of the inhomogeneous *g*- and *K*-functions that are generalizations of the Ohser estimator (see Wiegand and Moloney 2014: sections 3.1.2.6 and 3.1.2.7). Recall that the general univariate estimator of the pair correlation function is given by

$$\hat{g}(r) = \frac{1}{\hat{\lambda}^2}\frac{1}{2\pi r\ A}\sum_{i=1}^{n}\sum_{j=1}^{n,\neq} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)w_{ij}(r) = \frac{1}{2\pi r\ A}\sum_{i=1}^{n}\sum_{j=1}^{n,\neq} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)\frac{w_{ij}(r)}{\hat{\lambda}\hat{\lambda}}$$

where $\hat{\lambda}$ is an estimator of the overall intensity of the pattern in the observation window, the double sum counts all pairs of points that have the approximate distance *r* (defined by a box kernel with ring width *dr*), and $w_{\mathrm{ij}}(r)$ is an edge correction weight of point pair i–j that attempts to compensate for the points j located in the rings (with radius *r* and width *dr* around point i) that are located outside the observation window. There are two main options for correcting this.

The first option, the Ripley edge correction, basically scales the number of points in incomplete rings by a factor being the area of an incomplete circle (with origin at point i that passes through point j), relative to the area of the complete circle. This produces in general a different weight for each i–j point pair. To extend this weight to a certain class of inhomogeneous patterns with intensity function $\lambda(\mathbf{x})$, Baddeley et al. (2000) weighted each point i in the double sum by the value of the intensity function $\lambda(\mathbf{x}_\mathrm{i})$ at the location $\mathbf{x}_\mathrm{i}$ of point i. The resulting estimator is then given by

$$\hat{g}^{BMW}(r, \lambda(\mathbf{x})) = \frac{1}{2\pi r\ A}\sum_{i=1}^{n}\sum_{j=1}^{n,\neq} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)\frac{w_{ij}}{\lambda(\mathbf{x}_\mathrm{i})\lambda(\mathbf{x}_\mathrm{j})}$$

For homogeneous patterns we have $\lambda(\mathbf{x}) = \lambda$ and the homogeneous estimator given above is recovered. A problem of $\hat{g}^{BMW}$ that can seriously limit its application is that $\lambda(\mathbf{x}_\mathrm{i})$ should not be small at the locations $\mathbf{x}_\mathrm{i}$ of the points. *Programita* therefore uses estimators based on the second option not affected by this problem.

The second option of homogeneous edge correction, which is based on the **Ohser weights**, corrects only the expected final bias by estimating the mean area of rings with radius *r* and centers that are randomly distributed within the observation window. Thus, the weighting factor $w^{O}{}_{\mathrm{i,j}}$ is only a function of distance *r*, but not of individual point pairs i–j. The Ohser weight is given by $w^{O}(r) = A/\bar{\gamma}_W(r)$ where $\bar{\gamma}_W(r)$ is the so-called isotropized set covariance that can be estimated analytically for homogeneous patterns and simple shapes of the observation window.

The resulting Ohser estimator for homogeneous patterns is

$$\hat{g}^{O}(r) = \frac{1}{\hat{\lambda}_n^{2}} \frac{1}{2\pi\ rA} \sum_{i=1}^{n} \sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)[\frac{A}{\bar{\gamma}_W(r)}]$$

where $\hat{\lambda}_n = n / A$ is the natural estimator of the intensity. For inhomogeneous patterns (or irregularly shaped observation windows) a generalization of $\bar{\gamma}_W(r)$ can be obtained by means of simulations of a heterogeneous Poisson process with intensity function $\lambda(\mathbf{x})$:

$$\hat{g}^{O}_{\text{inhom}}(r, \lambda(\mathbf{x})) = \frac{1}{\hat{\lambda}_n^{2}} \frac{1}{2\pi rA} \sum_{i=1}^{n} \sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r) \frac{A}{\bar{\gamma}_W(r, \lambda(\mathbf{x}))} \qquad (1)$$

where $\hat{\lambda}_n = n / A$ is the natural estimator of the intensity. The denominator contains a generalized, isotropised set covariance

$$\bar{\gamma}_W(r, \lambda(\mathbf{x})) = \frac{1}{\hat{\lambda}_n^{2}} \frac{1}{2\pi r} \int_W \lambda(\mathbf{x}) \left[\int_W \lambda(\mathbf{a}) k(\|\mathbf{x} - \mathbf{a}\| - r) \mathbf{da}] \right] \mathbf{dx}$$

which needs to be estimated numerically by

$$\bar{\gamma}_W(r, \lambda(\mathbf{x})) \approx \frac{1}{\hat{\lambda}_m^{2}} \frac{1}{2\pi r} \sum_{i=1}^{m} (\sum_{j=1,\neq}^{m} k(\|\mathbf{y}_i - \mathbf{y}_j\| - r)$$

where the $\mathbf{y}_i$ are the points of an auxiliary heterogeneous Poisson process of intensity function $\lambda(\mathbf{x})$ that has a high number of points (= m) and $\hat{\lambda}_m = m / A$. With this approximation we find

$$\hat{g}^{O}_{\text{inhom}}(r, \lambda(\mathbf{x})) \approx \frac{(m-1)m}{(n-1)n} \frac{\sum_{i=1}^{n} \sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)}{\sum_{i=1}^{m} \sum_{j=1,\neq}^{m} k(\|\mathbf{y}_i - \mathbf{y}_j\| - r)} \qquad (2)$$

Note that the generalized isotropized set covariance $\bar{\gamma}_W(r, \lambda(\mathbf{x}))$ need to be calculated only once and can then be used for all simulations of the null model.

Programita offers also a version of this estimator based on adapted intensity estimators presented in Illian et al. (2008):

$$\hat{g}^{WM}_{\text{inhom}}(r, \lambda(\mathbf{x})) \approx \frac{m}{n-1} \frac{\sum_{i=1}^{n} \sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)}{\sum_{i=1}^{m} \sum_{j=1,\neq}^{m} k(\|\mathbf{x}_i - \mathbf{y}_j\| - r)} \qquad (3)$$

where only the second j–point in each point pair i–j is taken from the auxiliary pattern $\mathbf{y}_i$. This estimator therefore compares the relative number of points of the pattern in rings around the points $\mathbf{x}_i$ with the relative number of points of the auxiliary pattern in rings around the points $\mathbf{x}_i$. In contrast to the Ohser estimator, the double-sum in the denominator must be evaluated each simulation of the null model because the $\mathbf{x}_i$ change each simulation. This makes this estimator slower, even though it produces less variability.

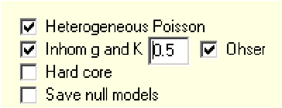
3.4.2 Inhomogeneous *g*- and *K* functions in *Programita*

One estimator of the inhomogeneous second-order summary functions is the inhomogeneous WM estimator detailed in equation 3.31 in Wiegand and Moloney (2014) and equation 3 above. This is an adapted estimator that conducts internally an estimation of a heterogeneous Poisson process based on the selected intensity function (the pattern $\mathbf{y}_j$ in equation 3). Thus, *Programita* simulates internally an auxiliary heterogeneous Poisson process for the estimation of the summary functions of both, the observed pattern and that of the simulated patterns. This makes the estimation somewhat slow. For this reason *Programita* uses the alternative Ohser estimator as default.

The number *m* of points of the auxiliary heterogeneous Poisson process used to estimate the expected area of rings or circles around the points of the pattern is proportionally to $n*m$ (adapted WM estimator) and proportionally to $m*(m - 1)$ (generalized Ohser estimator). To obtain reasonably quick numerical estimation of the inhomogeneous edge correction functions, *Programita* uses as default

$m = 6000*6000/n$ (adapted WM estimator)
$m = 30000$ (generalized Ohser estimator)

If you want to use more (or less) points than that of the default you can provide a factor in the Select a null model window left of the checkbox “Ohser” (0.5 in the example):



☑ Heterogeneous Poisson
☑ Inhom g and K 0.5 ☑ Ohser
□ Hard core
□ Save null models

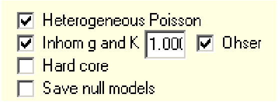
The temporary file temp_ec.dat shows the points of auxiliary heterogeneous Poisson process of the first simulation. Clearly, the more points you use the slower the estimation.

To speed up the estimation of the inhomogeneous summary functions *Programita* use the generalized Ohser estimator which is detailed in equation 3.29 and 3.30 in Wiegand and Moloney (2014) and in equation 2 above. This estimator is not adapted in a sense that it does not estimate (for irregularly shaped observation windows) the expected area of rings or circles around the actual points of the pattern (as the corresponding WM estimator), but it estimates a generalization of the isotropized set covariance (equation 3.30) that yields basically the expected area of rings or circles around the points of a heterogeneous Poisson process based on the intensity function selected. The advantage is that the generalized isotropized set covariance needs to be estimated only once (because it is independent on the actual locations of the points of the pattern), and once estimated it can be applied to both, the estimation of the inhomogeneous summary functions of the observed pattern and the simulated patterns.

The number of points of the auxiliary heterogeneous Poisson process for the generalized Ohser estimator ranges will be in general lower than that of the adapted WM estimator, with

$m = 30000$ (generalized Ohser estimator)

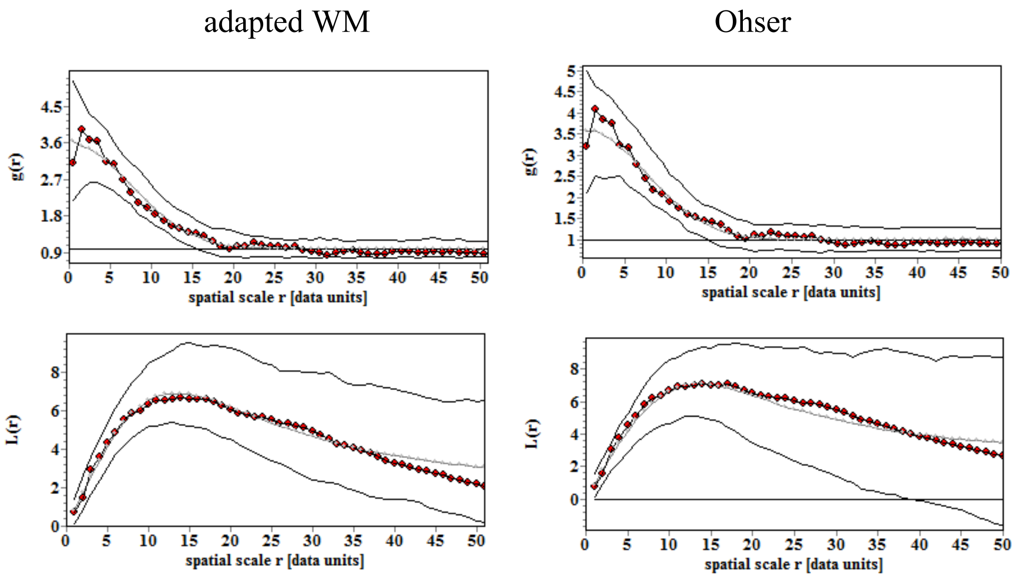
To use the inhomogeneous Ohser estimator click the check box “Ohser”:



☑ Heterogeneous Poisson
☑ Inhom g and K 1.00(☑ Ohser
□ Hard core
□ Save null models

.

Using the Ohser estimator in example Book_Fig4_19b_Ohser.res instead of the WM estimator (example Book_Fig4_19b) yields almost the same results (see below "Inhomogeneous Thomas process" for step-by-step instructions):


adapted WM

Ohser

You can also fit an inhomogeneous double cluster Thomas process; try for example the data set of Figure 3.13.

You can also use the generalized inhomogeneous double cluster Thomas process with negative Binomial distributions to govern the distribution of the number of points over the clusters of the underlying homogeneous point process.

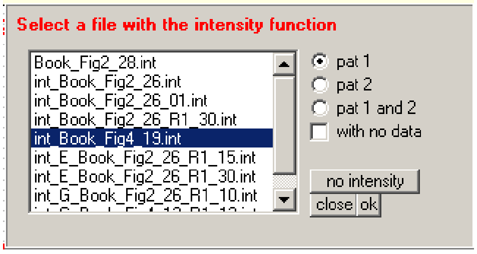
3.4.3 Variability in estimation of inhomogeneous summary functions

Because estimation of the inhomogeneous summary functions using the WM estimator uses an auxiliary (heterogeneous Poisson) pattern to estimate the edge correction it introduces a small stochastic variability. This variability can be easily assessed.

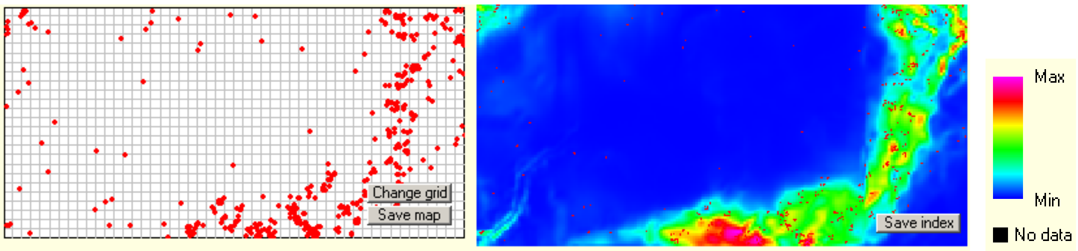
Example Book_Fig4_19_var.res

1. Execute *Programita*.
2. Highlight data file Book_Fig4_19b.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Enable check box "**Inhom g and K**" and disable "Ohser"

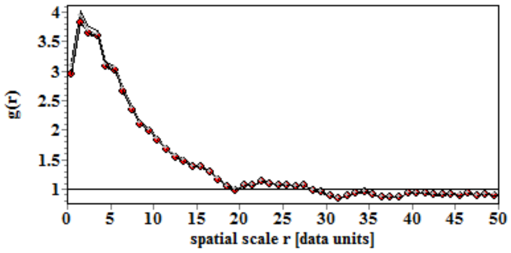
9. The window "Select a file with the intensity function" appears where you select the intensity file you want to use for estimation of the inhomogeneous second-order summary functions (i.e., int_Book_Fig4_19.int). Select "**pat 1**" (because it is the intensity of pattern 1) and then click the small "**ok**" button:

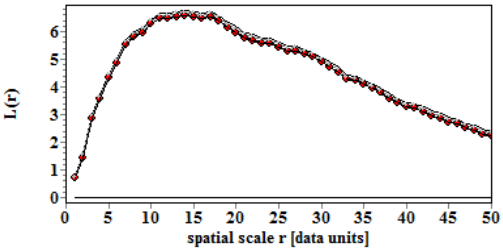

Select a file with the intensity function
Book_Fig2_28.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_Book_Fig2_26_R1_30.int
int_Book_Fig4_19.int
int_E_Book_Fig2_26_R1_15.int
int_E_Book_Fig2_26_R1_30.int
int_G_Book_Fig2_26_R1_10.int
● pat 1
○ pat 2
○ pat 1 and 2
□ with no data
no intensity
close
ok

Programita now shows the intensity function and the pattern:



10. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
11. Select in the window Select a null model "**Pattern 1 fix, pattern 2 CSR**". This null model does therefore estimate 199 times the summary functions of the observed pattern, thereby assessing the variability in the estimators of the inhomogeneous second-order summary functions.
12. Click "**Calculate Index**". *Programita* now estimates the inhomogeneous summary functions based on an auxiliary heterogeneous Poisson pattern with a default of m = 6000*6000/n_1 = 6000*6000/428 = 84,112 points. One example file with the auxiliary pattern is saved as temp_ec.dat.
13. The variability in the estimators of the inhomogeneous second-order summary functions is extremely small:

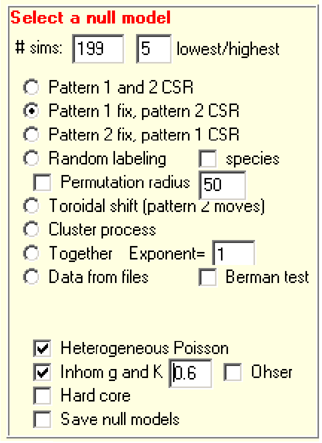




Example Book_Fig4_19_var06.res

To investigate how the variability in the estimators of the inhomogeneous second-order summary functions depends on the number of points of the auxiliary pattern change the number of points in the Select a null model window by giving a factor that increases/decreases the number of points. In the example we use the minimum (50,000 auxiliary points) which yields a factor 0.6.

First, with 50,000 auxiliary points the variability in the estimators of the inhomogeneous second-order summary functions is low:

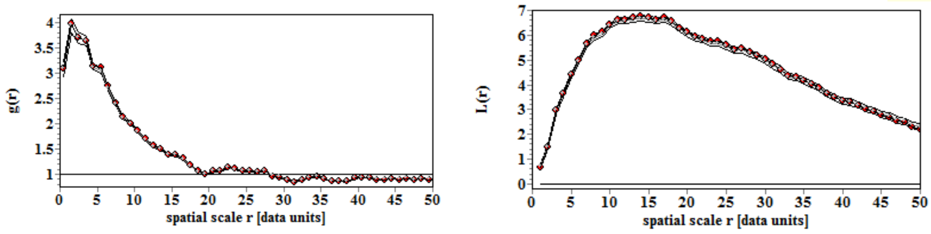


Select a null model

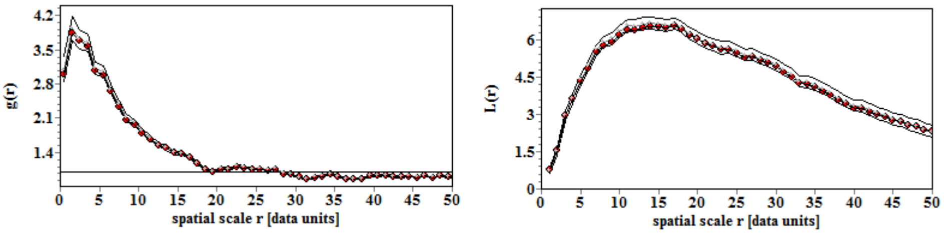
sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
● Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1
○ Data from files □ Berman test

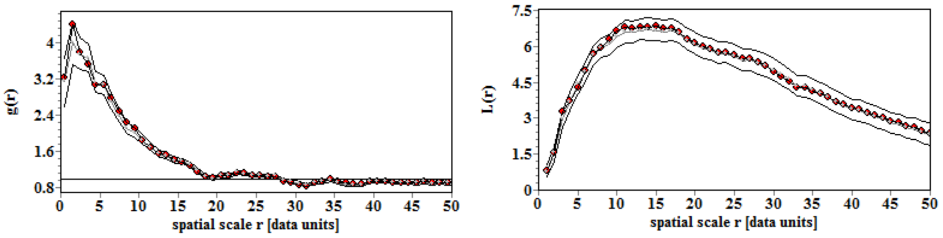
☑ Heterogeneous Poisson
☑ Inhom g and K 0.6 □ Ohser
□ Hard core
□ Save null models



Second, with 15,000 auxiliary points (factor 0.178) the variability in the estimators of the inhomogeneous second-order summary functions becomes notable:



Third, when using 4,200 auxiliary points (factor 0.05) the variability in the estimators of the inhomogeneous second-order summary functions becomes unacceptably large:

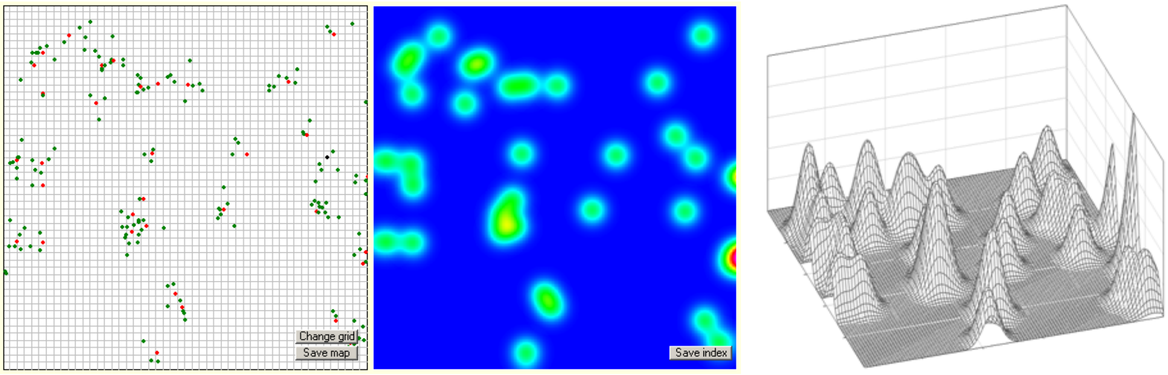


When using the WM estimator for inhomogeneous second-order summary functions it is a good idea to first check if the number of auxiliary points provides a good compromise between estimation speed and variability.

3.5 Cox processes and inhomogeneous cluster processes

Cox processes are a broad class of point processes that encompass and can combine heterogeneous Poisson processes and cluster processes. They are based on a generalized intensity function. More detail can be found in section 4.1.5 “Cox Processes” in Wiegand and Moloney (2014) and especially in Section 6.4 of the book of Illian et al. (2008, pp. 379–386).

For example, a simple Thomas process can also be generated with an intensity function that is based on superposition of two dimensional normal distributions that are centered on the locations of the cluster centers. Here is for example on the left the simple parent-offspring pattern Book_Fig4_13.dat shown in Figure 4.13 (see example Book_Fig4_13.res). The cluster centers are shown as red points and the points of the simple Thomas process as green points. The pattern was simulated based on 34 cluster centers and a parameter *σ* = 13.3 governing the normal distribution. In the middle and right is the intensity function that results from the superposition of the normal distribution with parameter σ = 13 centered on the 34 cluster centers (saved as file int_G_Book_Fig4_13_R1_13.int):



The Cox/simple Thomas process can be simulated in a straight forward way using the methods for the heterogeneous Poisson process. Points with random coordinates **x** are proposed and a point is accepted with a probability proportionally to $\lambda(\mathbf{x})/\lambda^*$where λ^* is the maximal value of $\lambda(\mathbf{x})$. These random trials are repeated as long as all *n* points of the pattern are distributed.

If the cluster centers are the same for all simulations of the point process, we have a parent-offspring Thomas process which is basically a heterogeneous Poisson process with a fixed intensity function $\lambda(\mathbf{x})$. However, we may also generate for each simulation of the point process a separate set of cluster centers as done in the simple Thomas process. This yields a “double stochastic” point process where the intensity function is itself stochastic and a realization of a stochastic process $\Lambda(\mathbf{x})$.

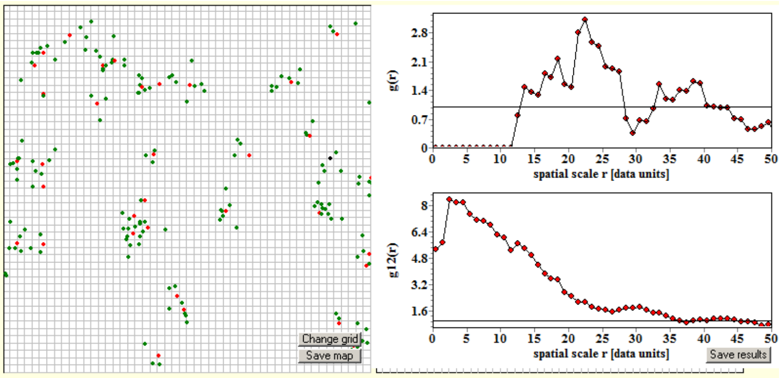
It is clear that superposition of an intensity function generated by homogeneous Thomas process with another intensity function that represents a large-scale trend, for example due to different habitat suitability (where the probability that a point is accepted depends on habitat suitability), generates an inhomogeneous Thomas process. This is an interesting feature of Cox processes that allows fitting inhomogeneous Thomas processes to the data.

Example Book_Fig4_18.res

In this example we reanalyze example Book_Fig4_13.res to illustrate the duality between a Cox process and a parent-offspring Thomas process where the cluster centers (i.e., parents) are known and are the same in each simulation of the point process.

The example file Book_Fig4_13.dat was generated with a simple Thomas process with parameter σ_S = 13.3, $A\ \lambda_1$ = 34 random clusters and 157 points and parameters.

1. Execute *Programita*.
2. Highlight data file Fig4_13.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 7 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. *Programita* shows the pattern, the univariate pair correlation function of the cluster centers and the bivariate pair correlation function of the points around their parents:



The pair correlation function of pattern 1 which is not of interest here is somewhat rugged because pattern 1 has few points (i.e., 34 clusters).

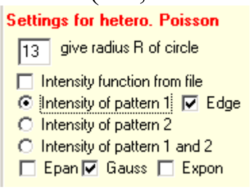
8. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.

11. Select in the window Select a null model "**Pattern 1 fix, pattern 2 CSR**".

12. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).

13. Click checkbox "**Heterogeneous Poisson**".

14. Go to window "Settings for hetero. Poisson" on the left and insert the bandwidth *R* (13m in the example) and enable "**Gauss**" for the Gaussian kernel. Now select "**Intensity of pattern 1**". In this case *Programita* estimates the intensity of pattern 1 (i.e., the cluster centers) based on a Gaussian kernel. (If you would select "Intensity of pattern 2" *Programita* would estimate the intensity of pattern 2). Edge correction "**Edge**" is enabled by default. *Programita* then uses this intensity function in the heterogeneous Poisson null model.



Settings for hetero. Poisson

13 give radius R of circle

□ Intensity function from file

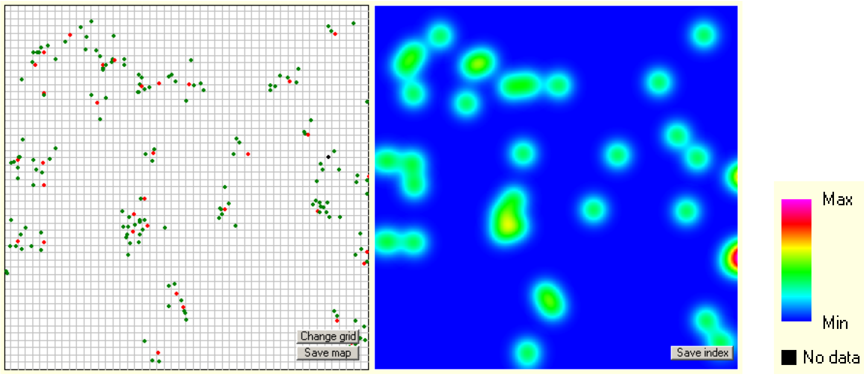
● Intensity of pattern 1 ☑ Edge

○ Intensity of pattern 2

○ Intensity of pattern 1 and 2

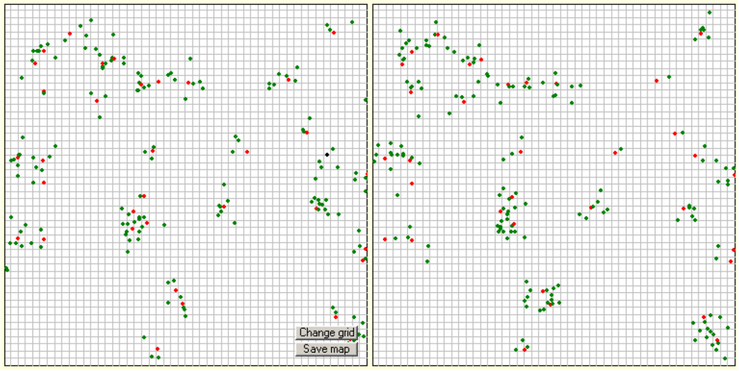
□ Epan ☑ Gauss □ Expon

15. Click “**Calculate Index**” and *Programita* estimates the intensity function and shows the pattern and the corresponding intensity function.

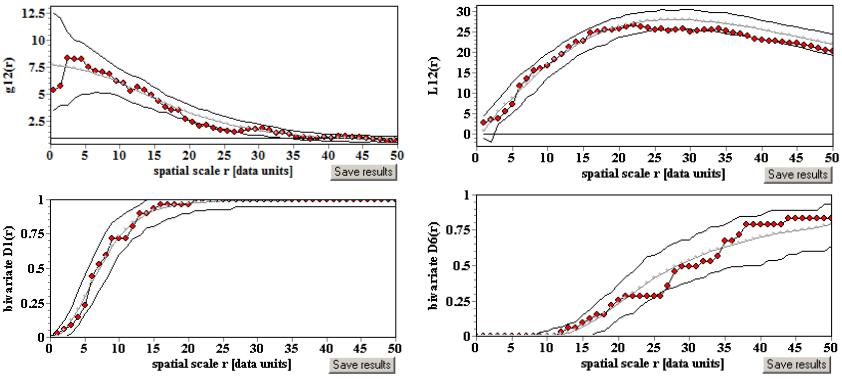


16. Click OK at the message box to save the intensity file. The file is saved with name int_G_Book_Fig4_13_R1_13.int where the “int_GE” indicates the Gaussian kernel, Book_Fig4_13.dat was the data file, “_R1_13” means that the intensity was estimated with pattern 1 and bandwidth 13.

    Now *Programita* conducts the analysis. You can observe during the simulations that the null model distributes the points of pattern 2 with probability proportionally to the intensity function. (If you would select e.g., null model “Pattern 1 and 2 CSR”, pattern 1 and 2 would be randomized following this intensity function) Here an example:



The result reproduce that of Fig. 4.18d-f and resembles that of example Book_Fig4_13bi.res well, outlining the equivalence of the Thomas and the Cox process:

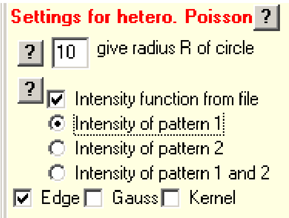


Example Book_Fig4_18_file.res

If you have the intensity file already saved, you can do the analysis also using this file. Starting with step 13 above:

13. Click checkbox “**Heterogeneous Poisson**”.
14. Go to window “Settings for hetero. Poisson”, select “**Intensity of pattern** 1” (because you use an estimate of the intensity of pattern 1) and then select “**Intensity file function from file**” on the left.

    The window “Select a file with the intensity function” appears where you can highlight the intensity file you want to use (i.e., **int_G_Book_Fig4_13_R1_13.int**). Select “**pat 1**” (because you use an estimate of the intensity of pattern 1), and then click the small **ok** button:



Settings for hetero. Poisson ?

? 10 give radius R of circle

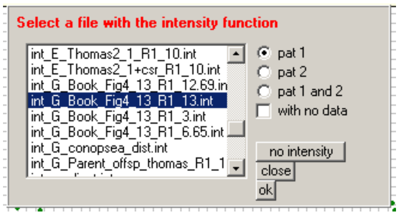
? ☑ Intensity function from file

● Intensity of pattern 1

○ Intensity of pattern 2

○ Intensity of pattern 1 and 2

☑ Edge □ Gauss □ Kernel



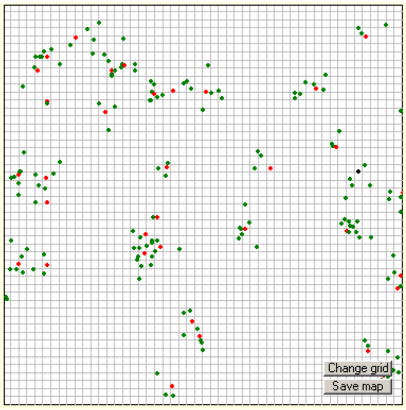
Select a file with the intensity function

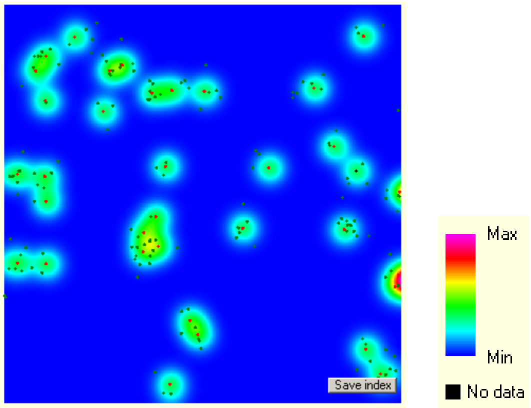
int_E_Thomas2_1_R1_10.int
int_E_Thomas2_1+csr_R1_10.int
int_G_Book_Fig4_13_R1_12.69.in
int_G_Book_Fig4_13_R1_13.int
int_G_Book_Fig4_13_R1_3.int
int_G_Book_Fig4_13_R1_6.65.int
int_G_conopsea_dist.int
int_G_Parent_offsp_thomas_R1_1

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

Programita then uses this intensity function in the selected null model. If your null model was “Pattern 1 and 2 CSR” the same intensity function is used for the heterogeneous Poisson process of pattern 1 and pattern 2, if your null model was “Pattern 1 fix and pattern 2 CSR” the intensity function is used for the heterogeneous Poisson process of pattern 2, and if your null model was “Pattern 2 fix and pattern 1 CSR” the intensity function is used for the heterogeneous Poisson process of pattern 1.

Programita shows the observed pattern (left) and the intensity function together with the data (right):



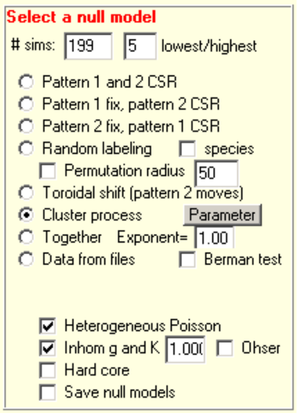


Click OK at the message box and “**Calculate Index**”. Now *Programita* conducts the same analysis as before in example Book_Fig4_18.res

3.5.1 Inhomogeneous Thomas process

We now fit an inhomogeneous Thomas process to the data shown in Figure 4.19b in Wiegand and Moloney (2014). The pattern is a realization of an inhomogeneous Thomas process based on a non-parametric estimation of the intensity function of all living individuals of the species *Ocotea whitei* from the 2000 census at the BCI plot (details on the habitat model can be found in Table 4.1 in Wiegand and Moloney 2014). The parameters of the underlying homogeneous Thomas process used to generate the pattern were σ = 4.8 and $A\rho$ = 581 cluster centers.

1. Execute *Programita*.
2. Highlight data file Book_Fig4_19b.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Enable check box "**Inhom g and K**" and disable "Ohser"
9. The window "Select a file with the intensity function" appears where you select the intensity file you want to use for estimation of the inhomogeneous second-order summary functions (i.e., int_Book_Fig4_19.int). Select "**pat 1**" (because it is the intensity of pattern 1) and then click the small "**ok**" button

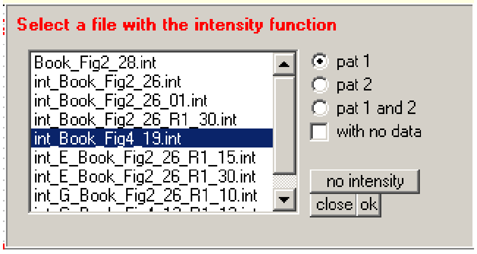


Select a null model

sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
● Cluster process Parameter
○ Together Exponent= 1.00
○ Data from files □ Berman test

☑ Heterogeneous Poisson
☑ Inhom g and K 1.00(□ Ohser
□ Hard core
□ Save null models



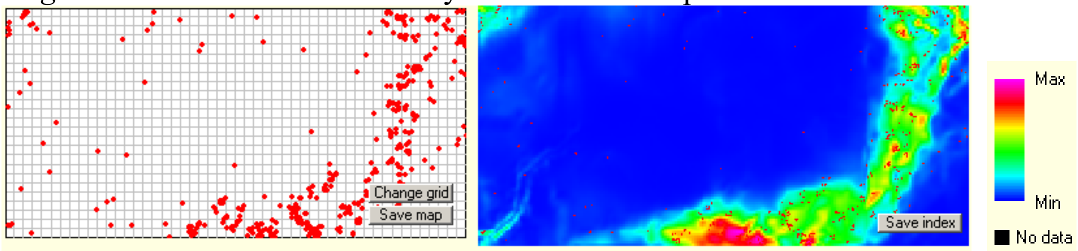
Select a file with the intensity function

Book_Fig2_28.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_Book_Fig2_26_R1_30.int
int_Book_Fig4_19.int
int_E_Book_Fig2_26_R1_15.int
int_E_Book_Fig2_26_R1_30.int
int_G_Book_Fig2_26_R1_10.int

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

no intensity
close ok

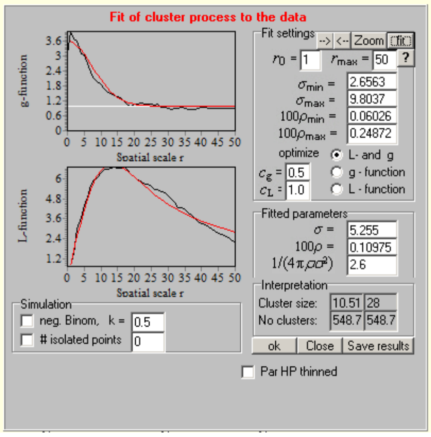
10. *Programita* now shows the intensity function and the pattern:



It is clear that most of the area is unsuitable (dark blue) and that many of the 581 clusters of the underlying homogeneous Thomas process will disappear.

11. To fit the inhomogeneous Thomas process to the data select "**Cluster process**" in the window "Select a null model".
12. A window "Fit of cluster process to data" opens. Select in the section "Null models" at the bottom "**Univar. cluster**". Continue with the small **ok** button.

13. Fit the parameters σ, and ρ. You obtain a good fit for σ =5.3, $A\rho$ = 548 which is close to the parameters σ = 4.8 and $A\rho$ = 581 used to generate the pattern. Note that one cannot expect a perfect agreement between the parameters used for generating the pattern and the fitted parameters of one realization. This is because the Thomas process is a stochastic process and because many of the original clusters of the underlying homogeneous Thomas process disappeared during thinning with the intensity function.



Fit of cluster process to the data

Fit settings

r_0 = 1 r_{max} = 50

σ_{min} = 2.6563

σ_{max} = 9.8037

$100\rho_{min}$ = 0.06026

$100\rho_{max}$ = 0.24872

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 5.255

100ρ = 0.10975

$1/(4\pi\rho\sigma^2)$ 2.6

Interpretation

Cluster size: 10.51 28

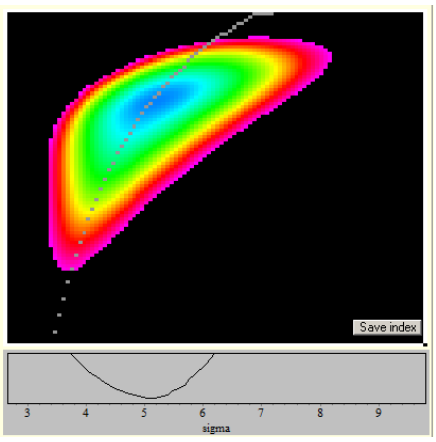
No clusters: 548.7 548.7

Simulation

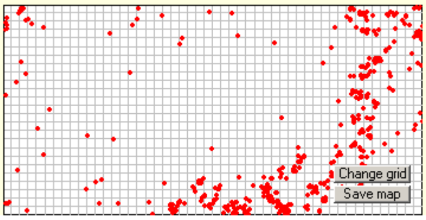
□ neg. Binom, k = 0.5

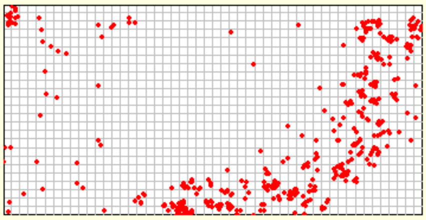
□ # isolated points 0

□ Par HP thinned

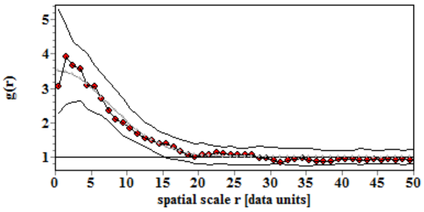


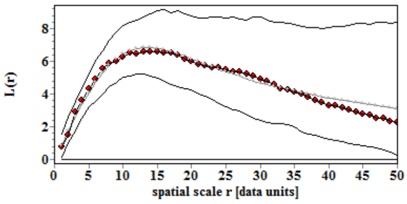
14. If you now click the "**ok**" and the "**Calculate Index**" you simulate the inhomogeneous Thomas process. As expected, the simulated pattern resembles the observed pattern well:



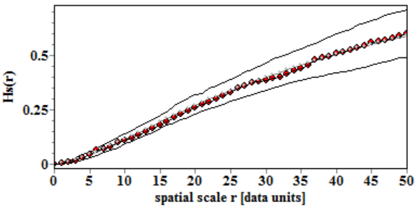


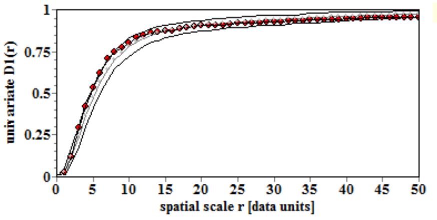
The pair correlation and L- function are well fitted:





the spherical contact distribution $H_s(r)$ and the distribution function $D^k(r)$ of the distances to the kth neighbor (here the 1th neighbor) as well (the GoF test for $D_1(r)$ yields a P-value of 0.19):





3.5.2 Series of analysis with inhomogeneous cluster point processes

The analyses of inhomogeneous cluster processes works exactly in the same way as for their homogeneous counterparts. However, in addition to the homogeneous cluster processes, *Programita* needs to read the file with the intensity function that contains the information on the environmental heterogeneity and is used to estimate the inhomogeneous second order summary factions.

Name convention for intensity files

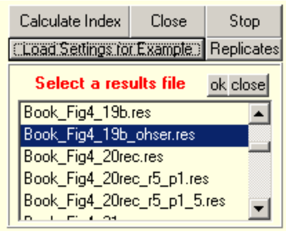
Programita needs to link the *.int intensity file to the corresponding *.dat data file. Therefore use the file list option to select the *.dat data files to be analyzed.

If a data file has name **Name.dat**, the intensity file must have the name **truncnameName.int** where truncname is an additional string common to all intensity files, suc as truncname=int_

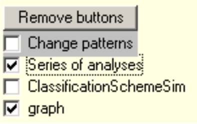
For example, if the data file is **Book_Fig4_19_1.dat**, the corresponding intensity file must be named **int_Book_Fig4_19_1**

Step-by-step example for series of analyses using inhomogeneous Thomas processes

The example uses point patterns generated by example Book_Fig4_19b_Ohser.res which generates patterns with driven by the intensity function int_Book_Fig4_19.int and a simple Thomas process with one scale of clustering with parameters (σ= 5.3500, ρ_L = 0.00103130). The data files are named Book_Fig4_19_n.dat.

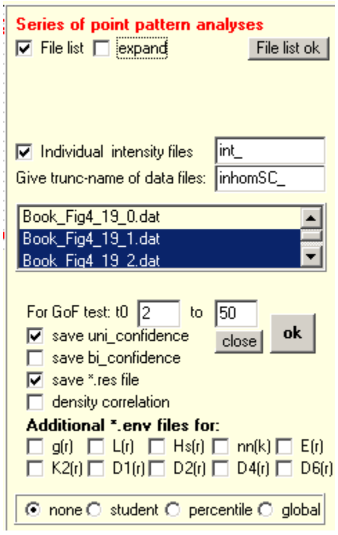


Book_Fig4_19b.res
Book_Fig4_19b_ohser.res
Book_Fig4_20rec.res
Book_Fig4_20rec_r5_p1.res
Book_Fig4_20rec_r5_p1_5.res



☐ Change patterns
☑ Series of analyses
☐ ClassificationSchemeSim
☑ graph

1. Read settings file Book_Fig4_19b_Ohser.res and run one simulation. Alternatively conduct one example analysis for the analyses you want to run as series.
2. Click "**Series of analyses**"
3. A window opens where you need to provide the specifications of your series of analysis. First click "**File list**" and "**expand**" to select the data files to be analyzed. Once all data files are highlighted select the truncname of the intensity files ("int_") and select a name for the analysis (e.g., inhomSC_"). The output results files will be named based on this name. Click the small button "File list ok".
4. Disable "save bi_confidence" because you run an univariate analysis
5. Provide distance interval (r_0, $r_{\max}$) for the GoF tests. In the example it is r_0 = 2 and $r_{\max}$ = 50.



☑ File list ☐ expand

☑ Individual intensity files: int_
Give trunc-name of data files: inhomSC_

Book_Fig4_19_0.dat
Book_Fig4_19_1.dat
Book_Fig4_19_2.dat

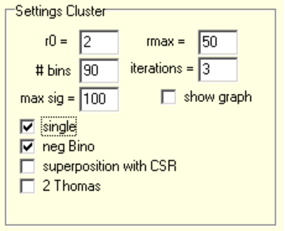
For GoF test: t0 2 to 50
☑ save uni_confidence
☐ save bi_confidence
☑ save *.res file
☐ density correlation

Additional *.env files for:
☐ g(r) ☐ L(r) ☐ Hs(r) ☐ nn(k) ☐ E(r)
☐ K2(r) ☐ D1(r) ☐ D2(r) ☐ D4(r) ☐ D6(r)

● none ○ student ○ percentile ○ global

6. Go to the box "**Settings cluster**" to select details of the cluster analysis. Please use the same cluster process as the *res file you used for running the example (i.e., Book_Fig4_19b_Ohser.res) or as used in the example analysis before running the series of analyses. Thus, if this was a single cluster process, enable "**single**". If it was a cluster process with two scales of clustering do not enable "**single**".

7. Provide distance interval (r_0, $r_{\max}$) for the fitting. In the example it is $r_0 = 2$ and $r_{\max} = 50$ (the same as for the GoF tests).



Settings Cluster

r0 = 2   rmax = 50

bins 90   iterations = 3

max sig = 100   □ show graph

☑ single

☑ neg Bino

□ superposition with CSR

□ 2 Thomas

8. Select "**# bins**", the number of intervals the interval of the parameters is derived into. *Programita* tests the entire grid of the (σ, ρ) parameter space (i.e., [# bins]^2 values are tested). The large the value the more precise is the estimate of the parameters but the longer the time requirement which increases with [# bins]^2. Select for example 90 bins.

9. Select "**max sig**" which gives the maximal value for parameter σ (100 in the example).

10. Select "**iterations**". This value determines the number of "zooms" into (or out of) the parameter space to get fines parameter estimates. A value of 3 iterations produces good results, if you select more the procedure is slower.

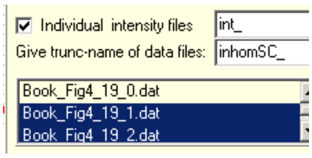
11. Select "**single**" only if your example is single cluster process.

12. If you select "**neg Bino**", *Programita* will first fit the standard Thomas process with two scales of clustering to the data (or with one scale of clustering if "single was selected") and then simulate the patterns for ten different values of the parameters k_L and k_S that govern how the points are distributed over the large and small clusters, respectively. We use here values of $k_L = k_S = 1000, 5, 1, 0.3, 0.1, 0.05, 0.03, 0.015, 0.0083, 0.0045, 0.0025, 0.0014$. Other options (e.g., $k_L = 1000$, and k_S variable) can be implemented if needed. This option works also together with the option "**single**".

13. If you select "**superposition with CSR**" *Programita* will first fit the standard Thomas process with two scales of clustering to the data (or with one scale of clustering if "single was selected") and then simulate the patterns for ten proportions of points being part of a CSR component pattern. The maximal percentage of CSR points can be selected with "**max. percent CSR points**".

14. If you select "**2 Thomas**" *Programita* will first fit the standard Thomas process with two scales of clustering to the data (or with one scale of clustering if "single was selected") and then simulate the patterns of the corresponding superposition of two Thomas processes with one scale of clustering for ten different values of proportion of points being part of the large scale Thomas process.

15. If all settings are correct, click the large "**ok**" button and then "Calculate index". *Programita* now conducts a series of identical analyses based on the data files Book_Fig4_19_n.dat



☑ Individual intensity files   int_

Give trunc-name of data files: inhomSC_

Book_Fig4_19_0.dat

Book_Fig4_19_1.dat

Book_Fig4_19_2.dat

3.6 Point processes generating regular patterns

3.6.1 Hard core processes

Cluster processes show an elevated neighborhood density where the typical point has more neighbors nearby than expected under a CSR process. In contrast, point processes showing regularity (or hyperdispersion) have a reduced neighborhood density. In the extreme case there is a minimal distance $2r_0$ between two points. This corresponds in the simplest case to randomly distributed disks with radius r_0 which do not overlap (i.e., a hard core process).

The hard-core process has a pair correlation function

$$g(r) = \begin{cases} 0 & \text{for } r \le 2r_0 \\ 1 & \text{for } r > 2r_0 \end{cases}$$

which is zero for distances r smaller than the diameter $2r_0$ of the disk and one for larger distances.

Hard core processes are often too simple to characterize inhibition processes producing observed point patterns in ecology, and more complex Gibbs or Markov Point Processes are used that can consider interaction functions of different shape (see section 4.1.6.2 "Gibbs or Markov Point Processes" in Wiegand and Moloney (2014) or sections 3.6 and 6.5 in Illian et al (2008)). However, in general Gibbs or Markov Point Processes cannot be simulated in a straight forward way as cluster or heterogeneous Poisson processes. Instead, they are governed by the so-called location density function (a high-dimensional probability density function) which yields basically the likelihood of a given point configuration. Simulation of this point processes requires optimization techniques where points of an initial pattern are deleted and replaced by randomly drawn points, which are accepted if the new point configuration becomes more likely, given the location density function. Such fitting procedures are not too different from individual-based simulation models that are based on biological mechanisms instead of purely statistical considerations. Therefore *Programita* has not implemented this type of point processes.

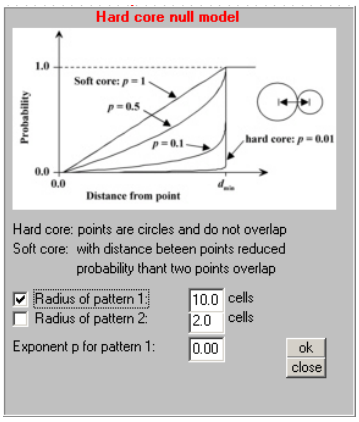
However, to provide you the possibility to simulate simple point patterns with hyperdispersion, *Programita* includes a simple algorithm to simulate a so-called "random sequential absorption" (RSA) process to produce simple hard core patterns.

The RSA algorithm implemented in *Programita* is simple. It is constructed by placing iteratively and randomly points within an observation window W which are thought to be the centers of disks with radius r_0. If a newly placed disk overlaps with an already accepted disk, it is not accepted, and new points are placed until the total number of points of the pattern is reached or until no further point can be placed because the pattern is "jammed".

Example Book_Fig2_11.res (RSA inhibition process)

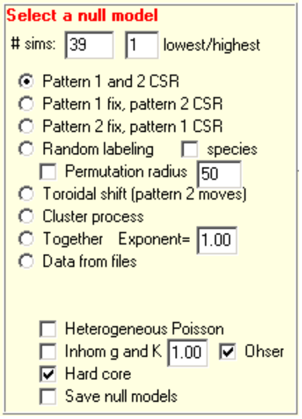
This pattern has been generated with a RSA algorithm to simulate non-overlapping disks with radius r_0 = 10m.

1. Execute *Programita*.
2. Highlight data file Book_Fig2_11.dat you want to analyze in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Click the button "change" in set maximal radius rmax to define the maximal scale r of the analysis and insert “100”
7. Press button “**Calculate Index**”
8. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
9. Select null model “Pattern 1 and 2 CSR” to start with the basic CSR algorithm.
10. Click checkbox “Hard core” and go to the window “Hard core null model” to define details of the RSA null model. Click “Radius of pattern 1” because you have a univariate pattern and provide the radius (10.0) in our case. To confirm settings click small “**ok”** button



Hard core: points are circles and do not overlap
Soft core: with distance beteen points reduced probability thant two points overlap

☑ Radius of pattern 1: 10.0 cells
□ Radius of pattern 2: 2.0 cells
Exponent p for pattern 1: 0.00

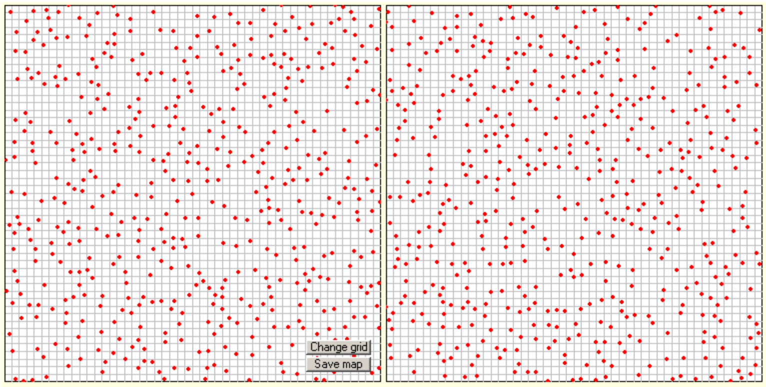


sims: 39 1 lowest/highest

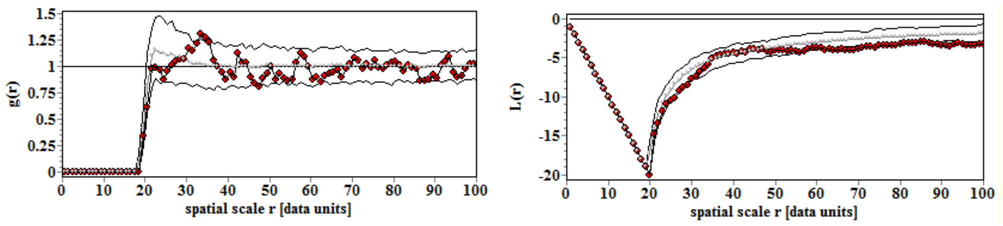
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
☑ Hard core
□ Save null models

11. To simulate the point process press “**Calculate Index**”. As expected, the simulated patterns look very similar to the observed pattern:

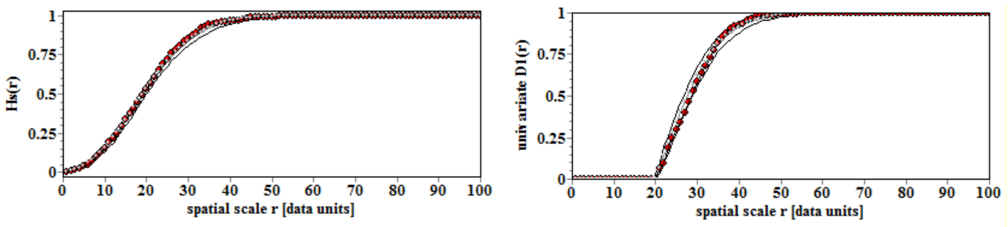


12. The pair correlation function and the *L*-function agree well with the simulated point process and show the typical “hard-core” shape:



Note however that the expectation of the RSA null model for the pair correlation function at distances slightly larger than the diameter $2r_0$ of the disk is not exactly one but somewhat larger. The reason for this is that in cases where already many points are placed the rejection rule causes acceptance of slightly more points close to an already placed point than farther away (because suitable gaps become scarce). As a consequence, we have a slight cluster effect.

13. The same is true for the spherical contact distribution and the nearest neighbor distribution function:

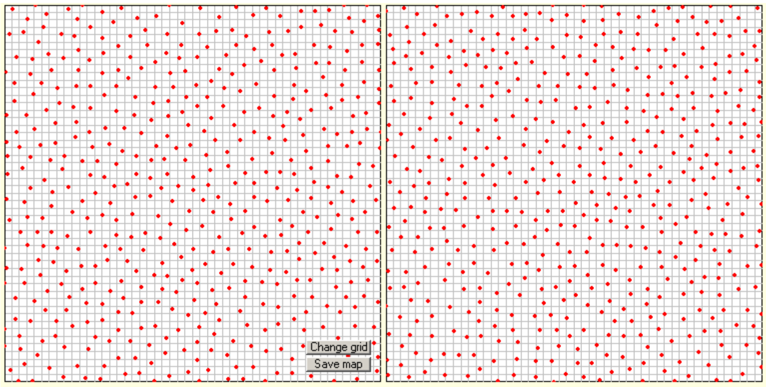


Example Book_Fig2_11jam.res (RSA inhibition process)

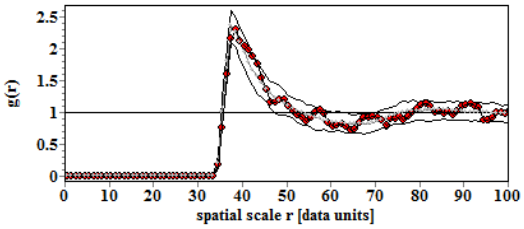
To see what happens if the pattern is close to jamming (i.e., no further points can be added) we increase the radius of the non-overlapping disks to r_0 = 18m. In this case the fraction of the observation 1000 × 1000 window covered by the disks yields $A_A = (500 \times \pi\ 18^2\ )/1000^2 = 0.509$. The maximum possible value of A_A for the RSA process yields $A_A = 0.547$ (which corresponds to a radius just below 19m). The file Book_Fig2_11_jam.dat is a realization of this point process with 500 points.

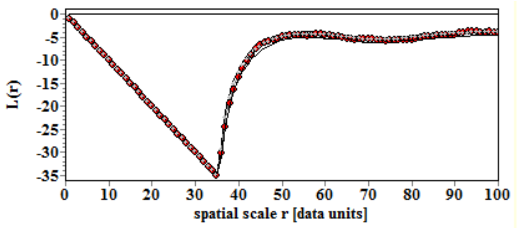
1. Execute *Programita*.
2. Highlight data file Book_Fig2_11jam.dat you want to analyze in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Click the button "change" in set maximal radius rmax to define the maximal scale *r* of the analysis and insert “100”
7. Press button “**Calculate Index**”

8. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
9. Select null model “Pattern 1 and 2 CSR” to start with the basic CSR algorithm.
10. Click checkbox “Hard core” and go to the window “Hard core null model” to define details of the RSA null model. Click “Radius of pattern 1” because you have a univariate pattern and provide the radius (18.0) in our case. To confirm settings click small **“ok”** button.
11. To simulate the point process press “**Calculate Index**”. You notice that the pattern is close to jamming because the simulation takes more time. This is because the algorithm needs many attempts to find a place for the last points. The resulting patterns are very regular patterns that almost yield a regular grid:

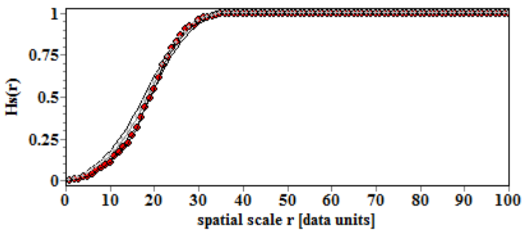


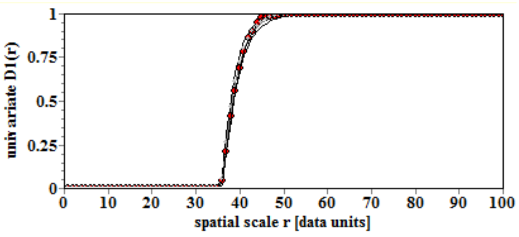
12. The pair correlation function at distances larger than the diameter $2r_0$ of the disk is now clearly elevated, it yields at the distance r = 40m a value of $g(r) \approx 2$ and then declines with one oscillation to the expected value of one. The closer the pattern to the jamming point, the higher the peak at distance $2r_0$. It is clear that point processes with the hard core pair correlation function shown above need more refined simulation methods:





The distribution function of the distances to the nearest neighbor switches over a very narrow range between r = 36 and r = 46 from zero to one. That means that all points have approximately the same distance to the nearest neighbor:

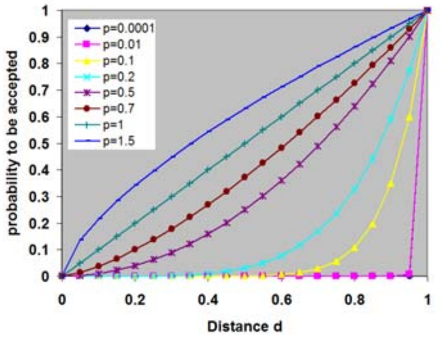




3.6.2 Soft-core processes

A soft core pattern arises if the radiuses of the disks are not the same or if the disks have a certain probability to overlap that depends on the distance to the nearest neighbor. *Programita* extends the sequential RSA process explained above where all radiuses are the same to a simple point process that yields a soft-core pattern. *Programita* uses a probability p_{HC} of a provisional point to be accepted that varies between 0 and 1, depending on the distance d to the nearest (accepted) neighbor, and an exponent p that gives the degree of “softness”:

$$p_{\mathrm{HC}}(d) \quad = \quad \begin{cases} d^{1/p} & \text{for } d \le 2r_0 \\ 1 & \text{for } d > 2r_0 \end{cases}$$

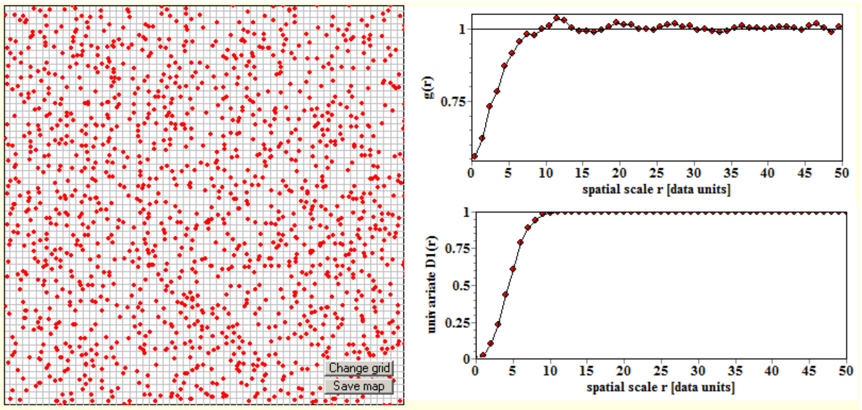


For $p = 0$, we obtain the RSA hardcore model, for $p > 0$ a soft core model, and for $p \rightarrow \infty$ we obtain CSR. The figure on the right shows how the rejection probability depends on the distance d to the nearest neighbor and the exponent p.

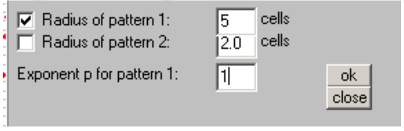
Example Book_Fig4_20rec.res (RSA soft core process)

Figure 4.20A in Wiegand and Moloney (2014) shows a 300 × 300m window from the BCI plot with all trees with a diameter larger than 20cm. The pair correlation function of this pattern shows a typical soft core shape (Fig. 4.20b). We analyze here instead a pattern with the same properties that was generated with pattern reconstruction based on the original BCI pattern for a 300 × 300m window.

1. Execute *Programita*.
2. Highlight data file Fig4_20rec.dat you want to analyze in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^{\mathrm{k}}(r)$.
6. Press button “**Calculate Index**”. As confirmed by the pair correlation function and the distribution function of the distances to the nearest neighbor, the pattern is a typical soft core pattern:



7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
8. Select null model “Pattern 1 and 2 CSR” to start with the basic CSR algorithm.
9. To fit the soft core process (manually) to the data click checkbox “**Hard core**” and go to the window “Hard core null model” to define details of the RSA null model. Click “**Radius of pattern 1**” because you have a univariate pattern. The pair correlation function and the distribution function of the distances to the nearest neighbor shown above suggest a maximum diameter of the disks of 10m, thus use as first estimate of the radius of pattern 1 a value of 5m. Because the pattern is quite soft, start with an exponent of p = 1.



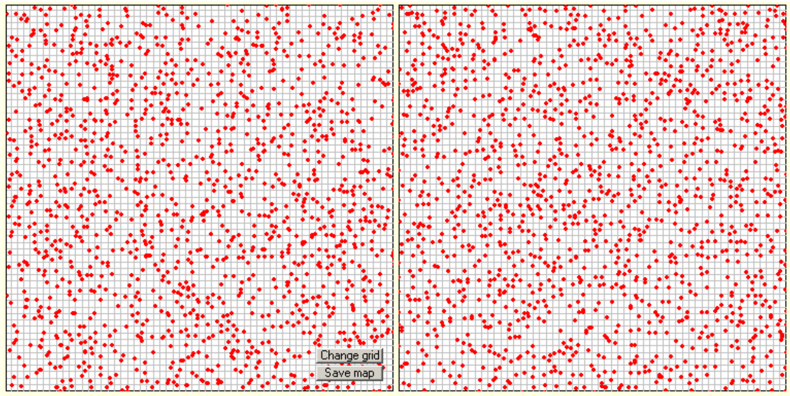
☑ Radius of pattern 1: 5 cells

□ Radius of pattern 2: 2.0 cells

Exponent p for pattern 1: 1

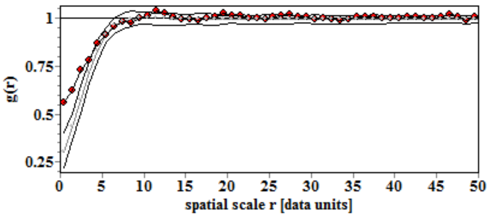
To confirm settings click small **“ok”** button.

10. To simulate the point process press “**Calculate Index**”. The simulated patterns look very similar to the observed pattern:

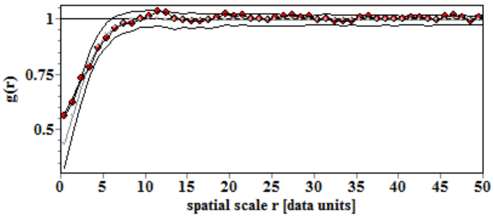


11. However, the pair correlation function is not well fitted at small distances, the observed pattern is softer. Therefore, use an exponent of say $p = 1.5$:

$r_0 = 5\text{m}, p = 1$

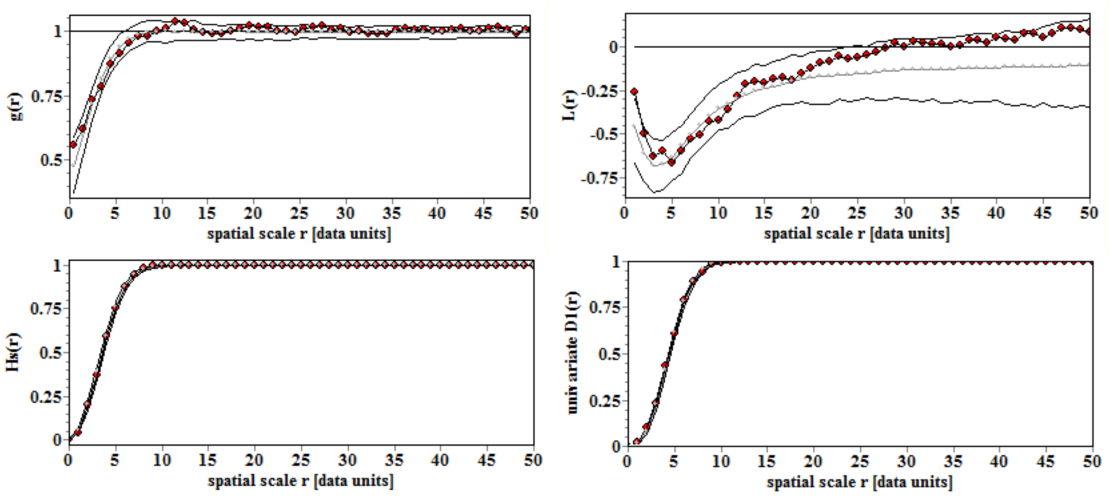


$r_0 = 5\text{m}, p = 1.5$



12. An exponent p= 1.5 still yields a small underestimation of the pair correlation function at small distances. Select now $p = 1.7$

13. The soft core RSA point process with parameters r_0 = 5m and p = 1.7 provides a good fit for all of the important summary functions:



The simulation confirmed that the trees with diameter larger than 20cm at BCI have a type of “zone of influence” of 5m (the radius r_0 = 5m) and the probability that a tree is inside this zone of influence of another tree declines almost with the square root of the distance to the focal tree.

3.7 Instructions for the Berman test in *Programita*

The Berman test can be used to test for a **significant association of a univariate point pattern to a continuous spatial covariate**. This test goes back to a study motivated by a geological problem (Berman 1986). The question was whether copper ore deposits (representing a point pattern) were spatially associated with linear-shaped geological features (lineaments) visible from satellite images. Berman (1986) reduced the problem to a test of spatial association between a point pattern and a spatial covariate $v(\mathbf{x})$. In the case of this motivating example, the spatial covariate $v(\mathbf{x})$ was the distance of a point located at **x** to the nearest lineament, but the test is applicable to any covariate $v(\mathbf{x})$.

The test is performed by comparing the observed distribution of the values of a spatial covariate $v(\mathbf{x})$ taken at the locations $\mathbf{x}_i$ of the points i of a point pattern and the predicted distribution of the same covariate under a null model that randomizes the points of the pattern.

The test statistic $Z_1 = (S - \mu)/\sigma$ introduced by Berman (1986) is based on the mean S of the covariate values $v(\mathbf{x}_i)$ at the points $\mathbf{x}_i$ of the observed pattern. The value μ is the predicted value of S under the null model and σ^2 the corresponding variance. The null distribution of this test statistic is approximately the standard normal distribution.

Based on this test statistic one can formulate the null and alternative hypotheses. The initial null hypothesis H_0 was that the pattern was generated by a homogeneous Poisson process (CSR) independent of the environmental covariate v(x). The alternative hypothesis H_1 was that the pattern is an inhomogeneous Poisson point process with an intensity function proportionally to the covariate $v(\mathbf{x})$. In a GoF test, significant deviation of the test statistic Z_1 from the null hypothesis H_0 can be assessed by comparing the observed value of Z_1 with the standard normal distribution or by comparing the rank of the observed S within the corresponding values of S for the simulations of the null model.

However, the original Berman test based on a homogeneous Poisson null model does **not take into account the effect of spatial autocorrelation** (clustering) in the point pattern and, therefore, it will detect too often a significant habitat association if the pattern is clustered (Berman 1986). Berman (1986) already proposed to use a **torus translation** null model to account for potential clustering of the pattern. A better alternative, however, is to use null distributions of the point pattern based on **pattern reconstruction** (Wiegand et al. 2013) that is able to generate stochastic replicates of the observed pattern that approximate several summary functions of the observed pattern very well (e.g., pair correlation function, *K*-function, spherical contact distribution, nearest neighbor distribution functions,...), thereby maintaining the observed spatial autocorrelation structure of the pattern very closely.

If null models other than CSR are used, the GoF test must be based on stochastic simulations of the null model. Here the values of μ and σ are estimated from the n simulations of the null model. The μ is the mean of the S values of the simulations of the null model and σ^2 the corresponding variance. The P-value of Z_1 can then be looked up from a table because Z_1 approximates the standard normal distribution. If $\boldsymbol{Z_1} < \mathbf{0}$ there is a **negative association** of the pattern with the covariate (because the observed value of S is smaller than the expected value μ) and if $\boldsymbol{Z_1} > \mathbf{0}$ the **association is positive**.

Note that the implementation of the Berman test in Spatstat is based on a (not documented) rank transformation of the covariate which however is not recommended in ecological applications. The rank transformation can change the P values of the test substantially. If you nevertheless want to use it, transform the covariate yourself and run the Berman test (in *Programita*) with the rank-transformed covariate instead of the original covariate.

The current implementation of the Berman test in *Programita* allows you to do both, use null models (e.g., the homogeneous Poisson process and the torus translation) implemented in *Programita* and to import null model files generated outside *Programita* (e.g., by the pattern reconstruction software). This provides the Berman test great flexibility to respond to your specific questions.

3.7.1 Input data for Berman test

This is the format required for point pattern data files. It is basically a list with the coordinates of the points of the pattern and a header containing information on the dimensions of the plot and the number of points:

```
0 250 0 250    169
0.30  9.40  1 0
0.60  11.601 0
3.80  13.701 0
6.40  8.90  1 0....
```

The first line gives the coordinates of the edges of the observation window xmin, xmax, ymin, xmax (0 250 0 250 in the example of a 250m × 250m plot) and the fifth number is the number of points of the pattern (169 in the example) and the following lines list the coordinates of the points. The file **must be space or tab delimited ASCII file with the *.dat extension**. If you use EXCEL for data preparation save the data file as tab delimited text file with name “name.dat”. The quotation marks force EXCEL to save your data as *.dat file.

The covariate must be stored as *.int file which again is a space or tab delimited ASCII file but based on an underlying discrete grid with the following structure:

```
1   250 1   250       62500  1.0000
   1    1 1   0.064804
   1    2 1   0.061006
   1    3 1   0.057456
   1    4 1   0.054125
…
```

The intensity function must use a grid **which must exactly match the dimensions of the plot.** For example, for a 250m × 250m plot one may use a grid of 1m × 1m cells which yields a 250 × 250 grid or 5m × 5m cells which yield a 50 × 50 grid. If you use a cells size other than 1 × 1 you must adjust this in *Programita* as well (i.e., the bin width, see below).

The header of the *.int data file gives the coordinates of the edges of the gridded observation window xmin, xmax, ymin, xmax (1 250 1 250 in the example of a 250m × 250m plot and 1m × 1m cells), the fifth number is the total number of cells (250*250 = 62500 in the example), and the last number is the cell size in units of the data (= 1m). Be careful that the bin width and the cell size fit together! For interpretation of the output note that *Programita* normalizes the intensity function in a way that the maximal value is 1.

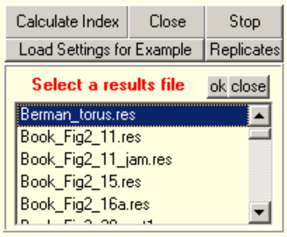
3.7.2 Name convention for null model file

The *Programita* implementation of the Berman test can use the toroidal shift and the homogeneous Poisson process (CSR) implemented in *Programita* as null model. However, *Programita* can also read the null model data previously generated by pattern reconstruction or any other null model that randomizes the observed pattern.

If you use the option to read the null model data from data files, they must meet simple name conventions. If the point pattern data file is for example called Berman.dat, the null model files must be called truncname_n.dat where the truncname is a string common to all null model files (in the example rec_Berman; pattern reconstruction assigns this name to reconstructions of the file Berman.dat) and n is the current number of the null model file, e.g., running from n = 1 to 199 if you use 199 realization of the null model for the Berman test.

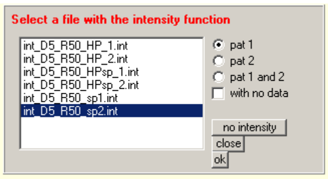
3.7.3 Running the Berman test, shortcut

The first step is to read the correct settings required for the Berman test. They are stored for example in the file Berman_torus.res that you can load when cliquing "**Load Settings for Example**" and "**ok**". *Programita* will then automatically select all settings that were used for this analysis.



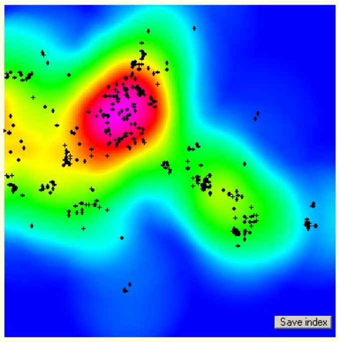
Berman_torus.res
Book_Fig2_11.res
Book_Fig2_11_jam.res
Book_Fig2_15.res
Book_Fig2_16a.res

First, a window "Select a file with the intensity function" appears where the covariate is selected. When using the example file to read the settings, the window shown on the right appears and the name of the covariate stored in the settings file is already highlighted (int_D5_R50_sp2.int) and "**pat1**" selected which means that the intensity of the first pattern is selected (i.e., an univariate analysis). Click "**ok**" and the covariate is shown together with the points of the pattern:

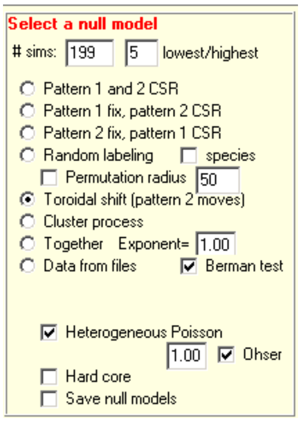


int_D5_R50_HP_1.int
int_D5_R50_HP_2.int
int_D5_R50_HPsp_1.int
int_D5_R50_HPsp_2.int
int_D5_R50_sp1.int
int_D5_R50_sp2.int

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data



Additional important settings automatically loaded are those associated with the selection of the null model in window "Select a null model". First, the checkbox "**Heterogeneous Poisson**" is checked (this is to read the covariate, not for the null model). Second, the checkbox "**Berman test**" must be checked, and the desired null model must be indicated ("Toroidal shift" in the example). Third, the number of **realizations of the null model** (199 in the example data) must be provided.

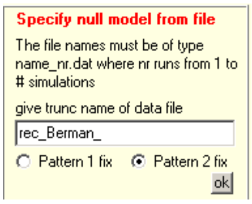


sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
● Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files ☑ Berman test

☑ Heterogeneous Poisson
1.00 ☑ Ohser
□ Hard core
□ Save null models

The example Berman_torus.res uses a null model implemented in *Programita*, an example for reading files with null model patterns is given in example Berman.res. Here 19 the files rec_Berman_1.dat, ..., rec_Berman_19.dat serves as example. You need to select the "**Data from file**" null model option to use these null model files. The window Specifiy null model from file opens where you have to provide details on the null model files; including the name conventions (i.e., the truncname "rec_Berman_" common to all null model files) explained above and selection of "**Pattern 2 fix**" (the latter indicates that pattern 1 is randomized). In the settings be sure that the number of realizations is the same as the number of files.



Specify null model from file

The file names must be of type name_nr.dat where nr runs from 1 to # simulations

give trunc name of data file

rec_Berman_

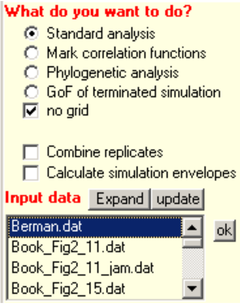
○ Pattern 1 fix ● Pattern 2 fix

ok

3.7.4 The Berman test with null model data files, step by step

To run the Berman test analysis, follow the following steps:

1. Select the **data file** (Berman.dat in the example)
2. How are your data organized: **List**
3. Select modus of data: "**List with coordinates, no list**"



What do you want to do?

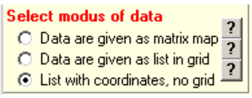
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

□ Combine replicates
□ Calculate simulation envelopes

Input data Expand update

Berman.dat
Book_Fig2_11.dat
Book_Fig2_11_jam.dat
Book_Fig2_15.dat

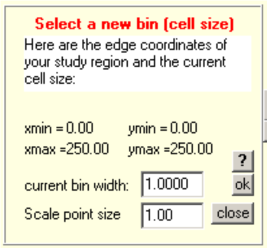
ok



Select modus of data

○ Data are given as matrix map ?
○ Data are given as list in grid ?
● List with coordinates, no grid ?

4. The window "Select a new bin (cell size)" appears. Select bin width to be 1 (default 1 unit). Note that you have to adapt the bin to the resolution (cell size) of the covariate data!



Select a new bin (cell size)

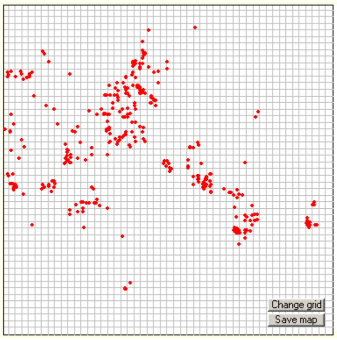
Here are the edge coordinates of your study region and the current cell size:

xmin = 0.00 ymin = 0.00
xmax =250.00 ymax =250.00

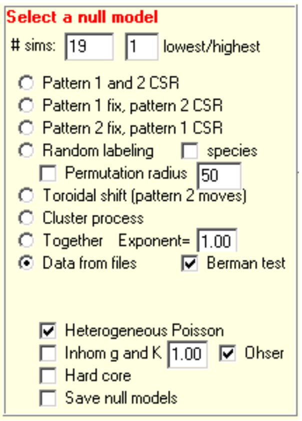
current bin width: 1.0000 ? ok

Scale point size 1.00 close

5. Click Calculate Index and *Programita* shows the observed pattern:



6. Click ☑ Calculate simulation envelopes , the widow "Select a null model" appears.
7. Be sure that the number of realizations is the same as the number of files. Therefore input 19 for # sims because the number of **realizations of the null model** is 19 in the example data. However, use for real analyses at least 199 null model patterns.



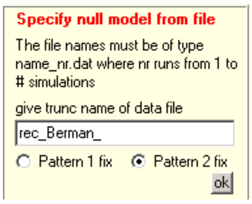
Select a null model

sims: 19 1 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
● Data from files ☑ Berman test

☑ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

8. Select "**Data from files**", "**Heterogeneous Poisson**", and "**Berman test**"
9. Specify names of files of realizations of null model ("rec_Berman_") and select "**Pattern 2 fix**" and then "ok".



Specify null model from file

The file names must be of type name_nr.dat where nr runs from 1 to # simulations

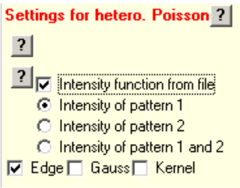
give trunc name of data file

rec_Berman_

○ Pattern 1 fix ● Pattern 2 fix

ok

10. Next, select in the window "Settings for hetero. Poisson" on the left the option "**Intensity function from file**"



Settings for hetero. Poisson ?

?

? ☑ Intensity function from file

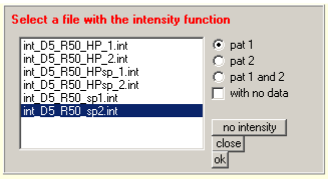
● Intensity of pattern 1

○ Intensity of pattern 2

○ Intensity of pattern 1 and 2

☑ Edge □ Gauss □ Kernel

11. A window "Select a file with intensity function" opens. Highlight the desired file with the intensity function (int_D5_R50_sp2.int) and clique "**ok**".



Select a file with the intensity function

int_D5_R50_HP_1.int
int_D5_R50_HP_2.int
int_D5_R50_HPsp_1.int
int_D5_R50_HPsp_2.int
int_D5_R50_sp1.int
int_D5_R50_sp2.int

● pat 1

○ pat 2

○ pat 1 and 2

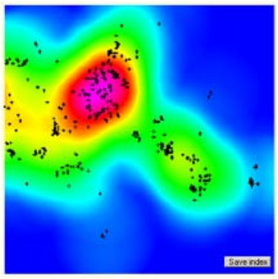
□ with no data

no intensity

close

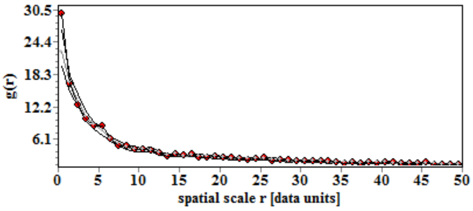
ok

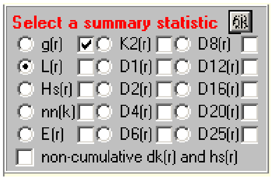
12. After cliquing ok, the covariate is shown together with the points of the pattern:



13. "Finally, click Calculate Index, now *Programita* reads the point pattern data file and the null model data files and conducts the Berman test.

14. *Programita* shows you the analysis of the univariate pattern to verify that the null model correctly conserved the observed univariate structure of the pattern:





Select a summary statistic ok

○ g(r) ☑ ○ K2(r) □ ○ D8(r) □

● L(r) □ ○ D1(r) □ ○ D12(r) □

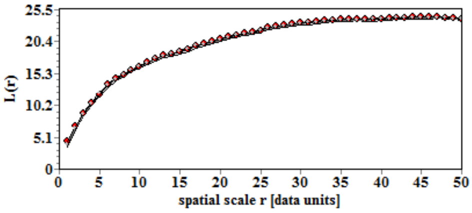
○ Hs(r) □ ○ D2(r) □ ○ D16(r) □

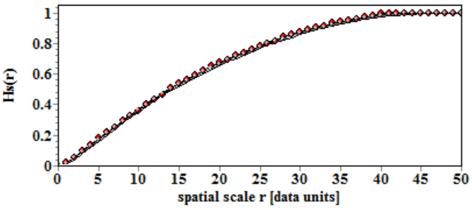
○ nn(k) □ ○ D4(r) □ ○ D20(r) □

○ E(r) □ ○ D6(r) □ ○ D25(r) □

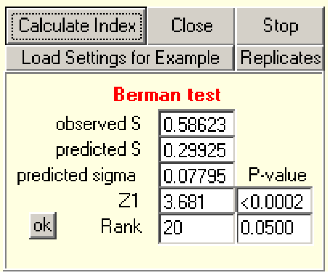
□ non-cumulative dk(r) and hs(r)

In this case the null model based on pattern reconstruction reproduced the structure of the univariate pattern very well. This can also be verified for other summary functions such as the L-function or the spherical contact distribution:





15. After the execution of the Berman test a box with the results appear that show
    - the **observed value of *S*** (being the mean of the covariate values $v(\mathbf{x}_i)$ at the points $\mathbf{x}_i$ of the pattern) (0.58623 in the example),
    - the predicted value of *S* under the null model = μ (i.e., the expected mean of the covariate values $v(\mathbf{x}_i)$ under a null model realization) estimated as the mean value over the *S* values resulting from the simulations of the null model (0.29925 in the example).
    - the predicted standard deviation σ of the values of *S* resulting from the null model realizations (0.07795 in the example),
    - the test statistic $Z_1 = (S - \mu)/\sigma$ (3.681 in the example). Z_1 is positive which indicates a positive association between the observed pattern and the covariate, and Z_1 is larger than 1.96 which indicates a moderate positive association.
    - the P-value of the test derived from the value of Z_1 (<0.002 in the example)
    - the rank of *S* within the corresponding values of *S* for the null model patterns where $P = 1 - (\text{rank-}1)/(\text{anzsim}+1)$.



Berman test

observed S 0.58623

predicted S 0.29925

predicted sigma 0.07795

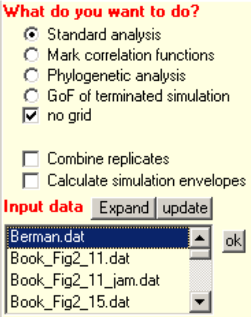
P-value

Z1 3.681 <0.0002

Rank 20 0.0500

3.7.5 Run the Berman test for CSR, step by step

To run the Berman test for null models implemented in *Programita* you have to conduct the following steps:



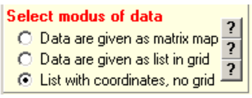
What do you want to do?

- ● Standard analysis
- ○ Mark correlation functions
- ○ Phylogenetic analysis
- ○ GoF of terminated simulation
- ☑ no grid
- □ Combine replicates
- □ Calculate simulation envelopes

Input data Expand update

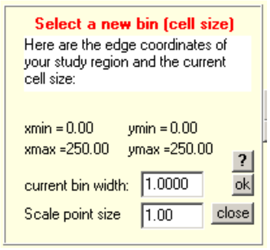
Berman.dat
Book_Fig2_11.dat
Book_Fig2_11_jam.dat
Book_Fig2_15.dat

1. Select the **data file** (Berman.dat in the example)
2. How are your data organized: **List**
3. Select modus of data: “**List with coordinates, no list**”
4. The window “Select a new bin (cell size)” appears. Select bin width to be 1 (default 1 unit). Note that you have to adapt the bin width to equal the resolution (cell size) of the covariate data!
5. Click Calculate Index and *Programita* shows the observed pattern:



Select modus of data

- ○ Data are given as matrix map
- ○ Data are given as list in grid
- ● List with coordinates, no grid



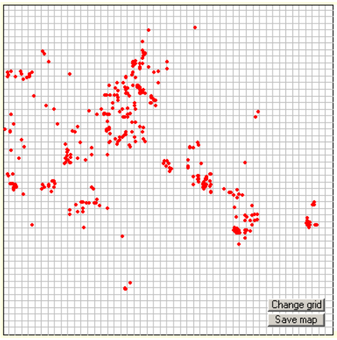
Select a new bin (cell size)

Here are the edge coordinates of your study region and the current cell size:

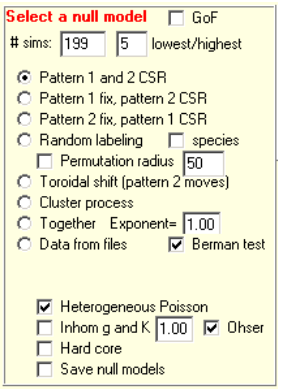
xmin = 0.00 ymin = 0.00
xmax =250.00 ymax =250.00

current bin width: 1.0000

Scale point size 1.00



6. Click ☑ Calculate simulation envelopes, the widow “Select a null model” appears.
7. Select “**Pattern 1 and 2 CSR**”.
8. The number of **realizations of the null model** (199 in the example data) must be provided.
9. Select “**Heterogeneous Poisson**” (to read the intensity file, not as null model!) and “**Berman test**”. A window “Settings for hetero. Poisson” opens on the left, select “**Intensity function from file**”

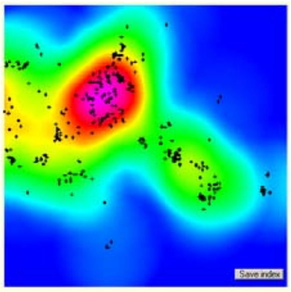


Select a null model □ GoF

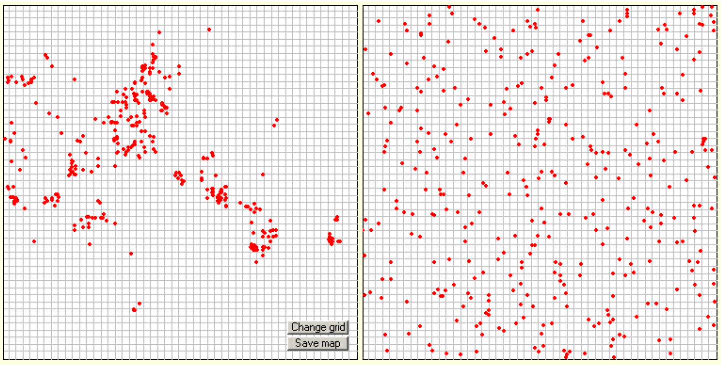
sims: 199 5 lowest/highest

- ● Pattern 1 and 2 CSR
- ○ Pattern 1 fix, pattern 2 CSR
- ○ Pattern 2 fix, pattern 1 CSR
- ○ Random labeling □ species
- □ Permutation radius 50
- ○ Toroidal shift (pattern 2 moves)
- ○ Cluster process
- ○ Together Exponent= 1.00
- ○ Data from files ☑ Berman test
- ☑ Heterogeneous Poisson
- □ Inhom g and K 1.00 ☑ Ohser
- □ Hard core
- □ Save null models

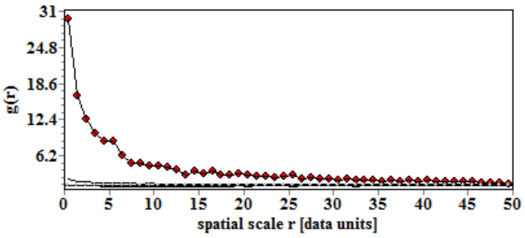
10. A window “Select a file with intensity function” opens. Highlight the desired file with the intensity function (int_D5_R50_sp2.int) and press “**ok**”, the covariate is shown together with the points of the pattern.
11. Be sure that “**Berman test**” is selected.
12. “Finally, click Calculate Index, now *Programita* reads the point pattern data file and conducts the Berman test based on the CSR null model.



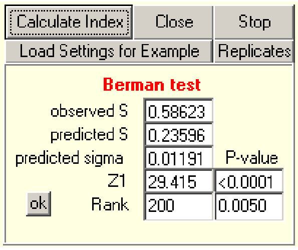
13. *Programita* shows on the left the observed pattern and ob the right the null model pattern:



14. After termination of the simulations of the null model *Programita* shows you the analysis of the univariate pattern to verify that the null model correctly conserved the observed univariate structure of the pattern:



In this case the null model based on CSR does not reproduce the observed cluster structure of the univariate pattern.



Berman test

observed S 0.58623

predicted S 0.23596

predicted sigma 0.01191 P-value

Z1 29.415 <0.0001

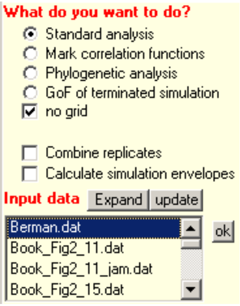
Rank 200 0.0050

15. After the execution of the Berman test a box with the results appear that show
    - the **observed value of *S*** (being the mean of the covariate values $v(\mathbf{x}_i)$ at the points $\mathbf{x}_i$ of the pattern) (0.58623 in the example),
    - the predicted value of S under the null model = μ (i.e., the expected mean of the covariate values $v(\mathbf{x}_i)$ under a null model realization) estimated as the mean value over the S values resulting from the simulations of the null model (0.23596 in the example. For comparison, μ = 0.33984 under the pattern reconstruction null model. These values should coincide for a homogeneous null model. The larger value of μ obtained in the pattern reconstruction null model is a consequence of the low number of simulations of the null model which does not yields a “representative” sample of the plot.
    - the predicted standard deviation σ of the values of S resulting from the null model realizations (0.01191 in the example) which is much lower than that of the pattern reconstruction null model (0.07408).
    - the test statistic $Z_1 = (S - \mu)/\sigma$ (29.415 in the example) which indicates a highly significant effect (compare with Z_1 = 3.326 for the pattern reconstruction null model). Z_1 is positive which indicates a positive association between the observed pattern and the covariate and Z_1 is larger than 1.96 which indicates a significant association.
    - Consequently, the P-value of the test derived from the value of Z_1 <0.0001 in the example)
    - the rank of S within the corresponding values of S for the null model patterns where P = 1 - (rank-1)/(anzsim+1).

3.7.6 Run the Berman test for toroidal shift, step by step

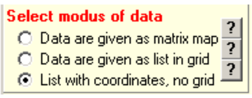
To run the Berman test analysis for null models implemented in *Programita* conduct the following steps.

1. Select the **data file** (Berman.dat in the example)
2. How are your data organized: **List**



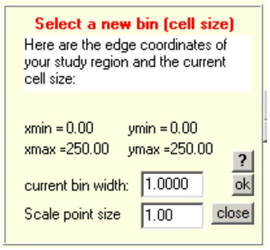
What do you want to do?
○ Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid
□ Combine replicates
□ Calculate simulation envelopes
Input data Expand update
Berman.dat
Book_Fig2_11.dat
Book_Fig2_11_jam.dat
Book_Fig2_15.dat
ok

3. Select modus of data: “**List with coordinates, no list**”



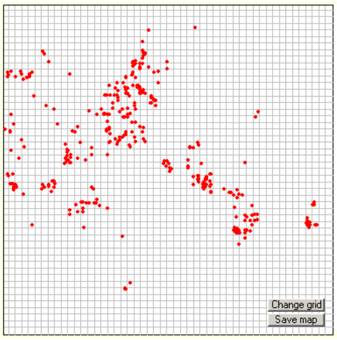
Select modus of data
○ Data are given as matrix map ?
○ Data are given as list in grid ?
● List with coordinates, no grid ?

4. The window “Select a new bin (cell size)” appears. Select bin width to be 1 (default 1 unit). Note that you have to adapt the bin width to equal the resolution (cell size) of the covariate data!

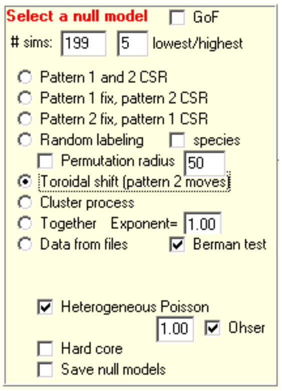


Select a new bin (cell size)
Here are the edge coordinates of your study region and the current cell size:
xmin = 0.00 ymin = 0.00
xmax =250.00 ymax =250.00
current bin width: 1.0000 ? ok
Scale point size 1.00 close

5. Click Calculate Index and *Programita* shows the observed pattern:

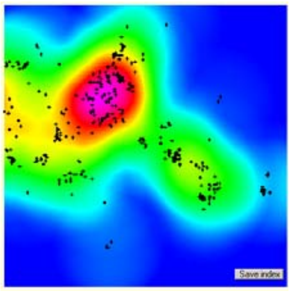


6. Click ☑ Calculate simulation envelopes, the widow “Select a null model” appears.
7. Select “**Toroidal shift**”.
8. The number of **realizations of the null model** (199 in the example data) must be provided.
9. Select “**Heterogeneous Poisson**” (to read the intensity file, not as null model!) and “**Berman test**”. A window “Settings for hetero. Poisson” opens on the left, select “**Intensity function from file**”



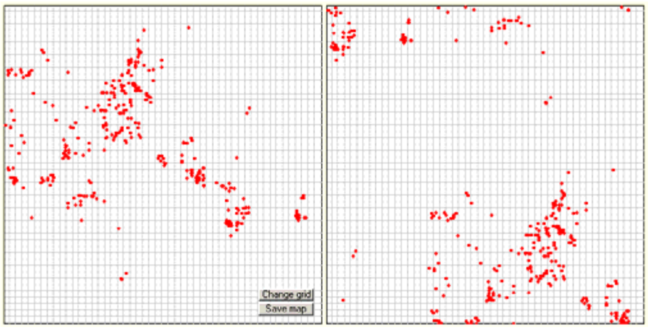
Select a null model □ GoF
# sims: 199 5 lowest/highest
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
● Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files ☑ Berman test
☑ Heterogeneous Poisson
1.00 ☑ Ohser
□ Hard core
□ Save null models

10. A window “Select a file with intensity function” opens. Highlight the desired file with the intensity function (int_D5_R50_sp2.int) and press “**ok**”, the covariate is shown together with the points of the pattern.

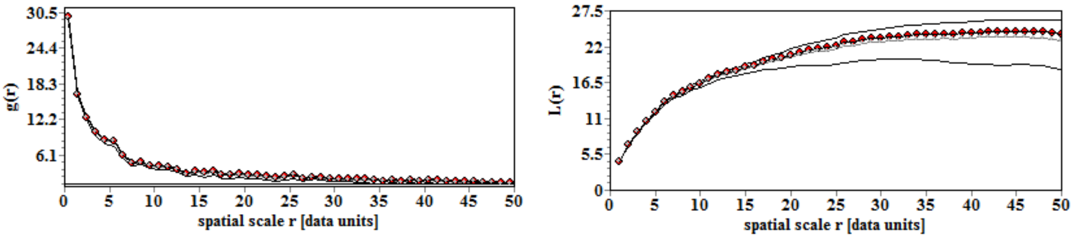


11. Be sure that “**Berman test**” is selected.
12. “Finally, click Calculate Index, now *Programita* reads the point pattern data file and conducts the Berman test with the toroidal shift null model.

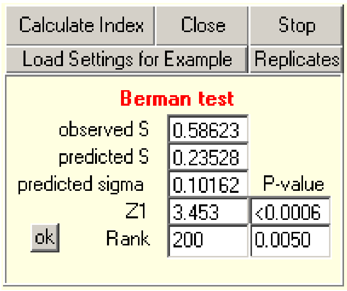
13. *Programita* shows on the left the observed pattern and ob the right the null model pattern:



14. After termination of the simulations of the null model *Programita* shows you the analysis of the univariate pattern to verify that the null model correctly conserved the observed univariate structure of the pattern:



In this case the torus shift null model reproduces, as expected, the observed cluster structure of the univariate pattern well.



Berman test

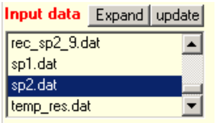
observed S 0.58623
predicted S 0.23528
predicted sigma 0.10162 P-value
Z1 3.453 <0.0006
Rank 200 0.0050

15. After the execution of the Berman test a box with the results appear that show
    - the **observed value of *S*** (being the mean of the covariate values $v(\mathbf{x}_i)$ at the points $\mathbf{x}_i$ of the pattern) (0.58623 in the example),
    - the predicted value of *S* under the null model = μ (i.e., the expected mean of the covariate values $v(\mathbf{x}_i)$ under a null model realization) estimated as the mean value over the *S* values resulting from the simulations of the null model (0.23528 in the example which coincides, as expected, with that of CSR).
    - the predicted standard deviation σ of the values of *S* resulting from the null model realizations (0.10162 in the example) which is larger than that of the pattern reconstruction null model (0.07408), which however used in the example only 19 simulations of the null model.
    - the test statistic $Z_1 = (S - \mu)/\sigma$ (3.453 in the example) which indicates a moderate significant effect (compare with $Z_1 = 27.557$ for CSR null model). Z_1 is positive which indicates a positive association between the observed pattern and the covariate and Z_1 is larger than 1.96 which indicates a significant association.
    - Consequently, the P-value of the test derived from the value of Z_1 <0.0001 in the example)
    - the rank of *S* within the corresponding values of *S* for the null model patterns where $P = 1 - (\text{rank-1})/(\text{anzsim+1})$.

3.7.7 Run the Berman tests for several data sets

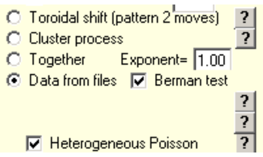
To run the Berman tests for several data sets you may need to change three settings:

1. The **point pattern data file**



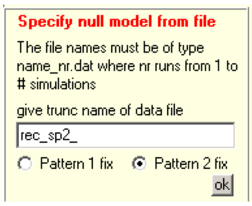
rec_sp2_9.dat
sp1.dat
sp2.dat
temp_res.dat

2. The **covariate data file**. This can be done by disabling and enabling the checkbox "Heterogeneous Poisson" in the null model selection window and then disabling and enabling the checkbox "Intensity function from file" in the "Settings for hetero. Poisson" window and then selecting the new intensity file.



○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
● Data from files ☑ Berman test
☑ Heterogeneous Poisson

3. Change the **name of the null model files** by cliquing "Toroidal shift" and "Data from file" in the " in the null model selection window and then update the trunc name of the null model files and clique "ok".



Specify null model from file

The file names must be of type name_nr.dat where nr runs from 1 to # simulations

give trunc name of data file

rec_sp2_

○ Pattern 1 fix ● Pattern 2 fix

Series of analyses to run the Berman tests for several data sets with their own covariates for toroidal shift or CSR

To run the Berman tests for several point pattern data with own covariate you can use the series of analysis option **with numbered files**.

You have a common name for all *.dat data files (in the example the truncname "HP_") and a number for each data file:
HPsp_1.dat
HPsp_2.dat
HPsp_3.dat
…

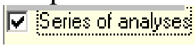
You have a common name for all *.int covariates (the truncname "int_D5_R50_HPsp_") that correspond to the *.dat pattern files:
int_D5_R50_HPsp_1.int
int_D5_R50_HPsp_2.int
int_D5_R50_HPsp_3.int

Note that it is better to use consistent name conventions for the *.dat pattern data files and the *.int intensity files (i.e., the name of the intensity file is composed of a common string (int_D5_R50_) followed by the name of the data files (HPsp_1).

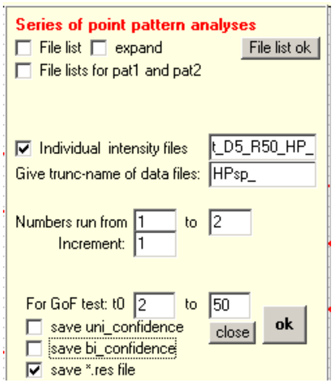
However, for the numbered file option this is not necessary. So you can also use intensity files int_D5_R50_sp1.int and int_D5_R50_sp2.int where the names of the data file and intensity file do not fully correspond, except the number.

Step by step

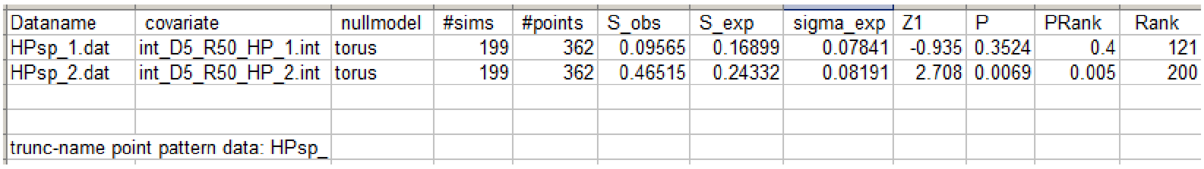
1. Read the settings of one analysis of the Berman test, e.g., the file BermanSeries_torus.res that uses
   - the data file HPsp_2.dat (truncname "HP_sp_"),
   - the covariate will be int_D5_R50_HPsp_2.int (truncname "int_D5_R50_HPsp_")
2. Run the example analysis with "**Calculate index**"
3. To run series of analysis the point pattern data must be named HPsp_1.dat, HPsp_2.dat, … and your covariates int_D5_R50_HPsp_1.int,.. int_D5_R50_HPsp_2.int,...
4. Clique "Series of analysis" on the right bottom

   
   ☑ Series of analyses

   and provide trunc-name of point pattern data files (= HPsp_) to yield data files HPsp_1.dat, HPsp_2.dat.
5. Click "Individual intensity files" and provide the truncname for the intensity files common to the intensity files for all species (= "int_D5_R50_HP_").
6. Provide the maximal number of data files (2), disable "**save uni_confidence**" and "**save uni_confidence**" and then "**ok**"


Series of point pattern analyses
□ File list □ expand [File list ok]
□ File lists for pat1 and pat2
☑ Individual intensity files [t_D5_R50_HP_]
Give trunc-name of data files: [HPsp_]
Numbers run from [1] to [2]
Increment: [1]
For GoF test: t0 [2] to [50]
□ save uni_confidence [close] [ok]
□ save bi_confidence
☑ save *.res file

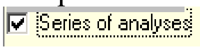
7. "**Calculate Index**". *Programita* now repeats the Berman test for the two data files HPsp_1.dat, and HPsp_2.dat based on toroidal shift null model.
8. *Programita* creates a (comma delimited) output file "Summary_BermanHPsp_.txt" that summarizes the results of the test for the different analyses. It also cerates the results files HPsp_1.res and HPsp_2.res
9. The file Summary_BermanHPsp_.txt:



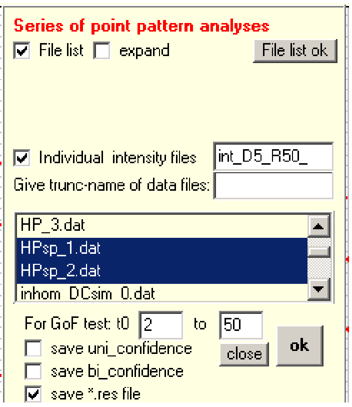
Dataname	covariate	nullmodel	#sims	#points	S_obs	S_exp	sigma_exp	Z1	P	PRank	Rank
HPsp_1.dat	int_D5_R50_HP_1.int	torus	199	362	0.09565	0.16899	0.07841	-0.935	0.3524	0.4	121
HPsp_2.dat	int_D5_R50_HP_2.int	torus	199	362	0.46515	0.24332	0.08191	2.708	0.0069	0.005	200
trunc-name point pattern data: HPsp_											

Alternatively, you can select the focal pattern to be analyzed that share a "trunc name" (i.e., HP_1.dat, HP_2.dat) **from a list of files**. To this end:

1. Conduct the Berman test analysis for one of the data files (e.g., with example BermanTorus.res).
2. Clique "Series of analysis"

   
   ☑ Series of analyses

   Click "Individual intensity files" and provide the truncname for the intensity files common to the intensity files for all species (= "int_D5_R50_").

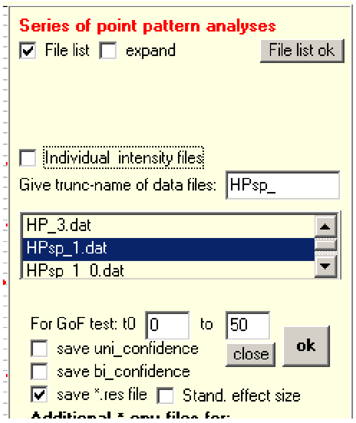

Series of point pattern analyses
☑ File list □ expand [File list ok]
☑ Individual intensity files [int_D5_R50_]
Give trunc-name of data files: [ ]
HP_3.dat
HPsp_1.dat
HPsp_2.dat
inhom_DCsim_0.dat
For GoF test: t0 [2] to [50]
□ save uni_confidence [close] [ok]
□ save bi_confidence
☑ save *.res file

3. Note that in this case you need exact correspondence between the data and intensity files (e.g., HPsp_1.dat, int_D5_R50_HPsp_1.int") because *Programita* must be able to reconstruct the name of the intensity file from the data name and the truncname (here "int_D5_R50_").
4. Click "File list" and "expand" and highlight the files you want to analyze (HPsp_1.dat and HPsp_2.dat). Then click "**File list ok**" and disable "**save uni_confidence**", "**save bi_confidence**" and "**ok**".
5. "Calculate Index". *Programita* now repeats the Berman test for the two data files HPsp_1.dat and HPsp_2.dat based on the toroidal shift null model.
6. *Programita* creates a (comma delimited) output file "Summary_BermanHPsp_.txt" that summarizes the results of the test for the different analyses. It also cerates the results files HPsp_1.res and HPsp_2.res

Series of analysis for several point pattern data sets with the same covariate

To run the Berman tests for several point pattern data sets with the same covariate proceed as in the cases above.

There is only one small difference, disable the option "Individual intensity function" and *Programita* will use for the analysis of all point pattern data files the same covariate.



Series of point pattern analyses

☑ File list □ expand

□ Individual intensity files

Give trunc-name of data files: HPsp_

HP_3.dat
HPsp_1.dat
HPsp_1_0.dat

For GoF test: t0 0 to 50

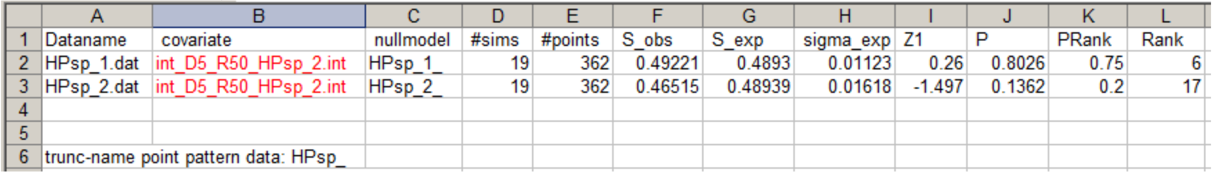
□ save uni_confidence

□ save bi_confidence

☑ save *.res file □ Stand. effect size

Additional *.env files for:

The file Summary_BermanHPsp_.txt



	A	B	C	D	E	F	G	H	I	J	K	L
1	Dataname	covariate	nullmodel	#sims	#points	S_obs	S_exp	sigma_exp	Z1	P	PRank	Rank
2	HPsp_1.dat	int_D5_R50_HPsp_2.int	HPsp_1_	19	362	0.49221	0.4893	0.01123	0.26	0.8026	0.75	6
3	HPsp_2.dat	int_D5_R50_HPsp_2.int	HPsp_2_	19	362	0.46515	0.48939	0.01618	-1.497	0.1362	0.2	17
4												
5												
6	trunc-name point pattern data: HPsp_											

4 Bivariate analysis in the standard mode

Bivariate analysis deals with a data type that comprises two types of points and is usually concerned with the characterization of the small-scale interaction structure between the two types of points although it can also be used to explore (larger-scale) co-occurrence patterns that may be influenced by habitat effects . A bivariate pattern is composed of two univariate component patterns which were created *a priori* by a different set of processes (e.g., two different species of trees in a forest). Chapter 4.2 in Wiegand and Moloney (2014) provides examples for the different analyses of bivariate patterns that are useful in ecology.

4.1 Getting started

4.1.1 Data preparation

Bivariate patterns comprise the coordinates of the two component point patterns. The data files for bivariate standard analysis must be an ASCII file with the *.dat extension and the following format (the example are the first lines of the file Book_Fig4_21a.dat):

```
0  200  0  200   500
 0.2   56.4  1  0
 0.4  133.6  0  1
 0.4  144.6  0  1
 0.8   19.4  1  0
 1.0   49.6  0  1
 2.4   52.2  1  0
 2.6  177.4  0  1
 3.0  123.8  1  0
 3.8   37.0  0  1
 4.6  196.0  1  0
 5.0   83.2  1  0
 5.0  146.2  0  1
```
…

where the first line gives the size of the observation window (200 × 200 units in the example) and the number of points in the pattern (= number of lines following the header). The first two columns are the coordinates, an entry “1” in the third column indicates that the point is of pattern 1 (i.e., a type 1 focal point) and an entry “1” in the fourth column indicates that the point is of pattern 2 (i.e., a type 2 point). The value of the third and the forth columns must be for the standard analysis mode “0 1” or “1 0”, no larger numbers or “1 1” are allowed.

The data file must be a space or tab delimited ASCII file with the *.dat extension. If you use Excel, there is a simple, but obviously generally unknown, way of saving files of a given type with a given extension:

1. Prepare the data file in Excel following the instructions above
2. Then save as a tab delimited text file, but write “name.dat” for the name (usually you would only write name and end up with a file named name.txt). The quotation marks are important because they force Excel to save the comma delimited file as name.dat.

4.1.2 Steps of bivariate analysis in standard mode

Programita estimates for data files of this type several summary functions based on estimators detailed in Illian et al. (2008) and Chapter 3 of Wiegand and Moloney (2014). The window Which method will you use allows you to specify the estimators.

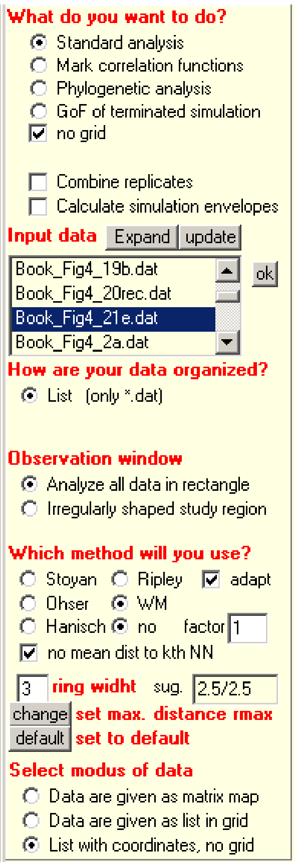
The standard analysis mode can be accessed with the following sequence of actions:

1. Select "**Standard analysis**" in window What do you want to do?
2. Highlight a data file in Input data ("Book_4_21e.dat" in the example) and click the small "ok" button.
3. The window Select a new cell size opens and allows you to provide a bin for your analysis given in units of your data. For example, if your data are in meter units and your observation window is 200 × 200m in size, an appropriate bin would be 1m. Press "**ok**" to confirm selection of the bin.

4. After selection of the bin ***Programita* suggests a ring width *dr*** based on equation 4.3.43 in Illian et al. (2008) [$dr = 0.2/\lambda^{0.5}$]. This equation provides a rough starting point for deciding on the ring width.

   The estimators of the pair correlation function implemented in the standard mode of *Programita* use a **default ring width of one bin** to obtain non-overlapping concentric rings. For reasons of computational efficiency you can then select only ring widths adding one bin in each direction, i.e., ring widths of 1, 3, 5, 7, … bins. You can change the ring width at the menu "Which method will you use". In the example file "Book_Fig4_21a.dat" with 250 type 2 points within a 200 × 200m observation window and a bin of 1m this yields a ring width of $dr = 2.5$. Thus select a ring width of 3.

5. Selecting the option "**no grid**" opens also a small window where you can select the desired rank *k* of the distribution functions $D^k(r)$ of the distances to the kth neighbor. Default is $k = 1, 2, 4, 6, 8, 12, 16, 20$, and 25. You can thus select the rank k of nine different functions $D^k(r)$. To confirm press the small **ok** button.



What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

☐ Combine replicates
☐ Calculate simulation envelopes

Input data

Book_Fig4_19b.dat
Book_Fig4_20rec.dat
Book_Fig4_21e.dat
Book_Fig4_2a.dat

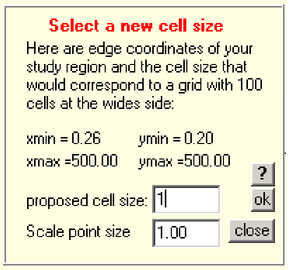
How are your data organized?
● List (only *.dat)

Observation window
● Analyze all data in rectangle
○ Irregularly shaped study region

Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1
☑ no mean dist to kth NN

3 ring widht sug. 2.5/2.5
set max. distance rmax
set to default

Select modus of data
○ Data are given as matrix map
○ Data are given as list in grid
● List with coordinates, no grid

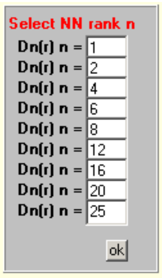


Select a new cell size

Here are edge coordinates of your study region and the cell size that would correspond to a grid with 100 cells at the wides side:

xmin = 0.26 ymin = 0.20
xmax =500.00 ymax =500.00

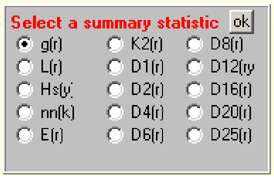
proposed cell size: 1
Scale point size 1.00



Select NN rank n
Dn(r) n = 1
Dn(r) n = 2
Dn(r) n = 4
Dn(r) n = 6
Dn(r) n = 8
Dn(r) n = 12
Dn(r) n = 16
Dn(r) n = 20
Dn(r) n = 25

6. Press button “**Calculate Index**” and *Programita* estimates a variety of summary functions of the uni- and bivariate data:

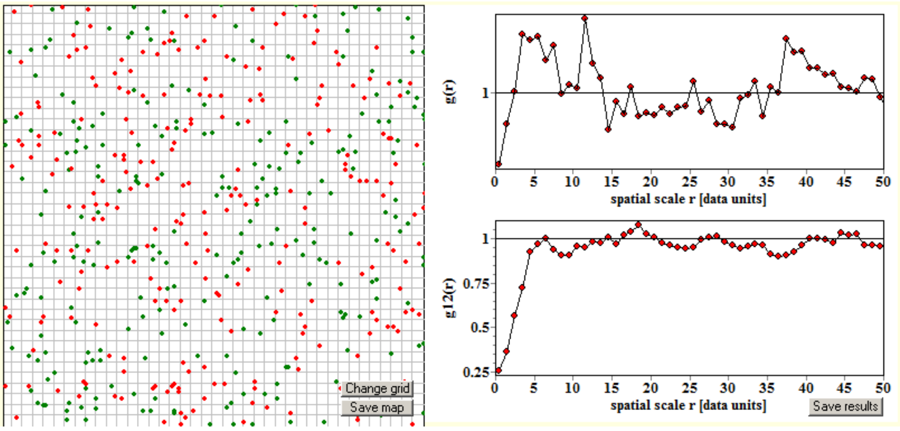
    - $g(r)$: pair correlation function
    - $L(r)$: L-function,
    - $H_s(r)$: the spherical contact distribution (only univariate)
    - $nn(k)$ the mean distance to the kth neighbor
    - $E(r)$ the probability that a point has no neighbor at distance within distances (r - 0.5, r + 0.5)
    - $K2(r)$ the $K2$ function
    - $D^k(r)$, the kth nn distribution functions, here with k = 1, 2, 4, 6, 8, 12, 16, 20, and 25



Select a summary statistic [ok]

● g(r) ○ K2(r) ○ D8(r)
○ L(r) ○ D1(r) ○ D12(r)
○ Hs(y) ○ D2(r) ○ D16(r)
○ nn(k) ○ D4(r) ○ D20(r)
○ E(r) ○ D6(r) ○ D25(r)

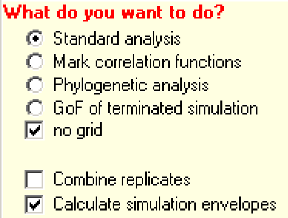
The screen shows on the left the bivariate pattern Book_Fig4_21e.dat with type 1 points (red) and type 2 points (green). On the right shown are the selected summary functions. The graph for the (partial) univariate summary functions (i.e., the analysis of only type 1 points) is shown on the top, and the bivariate summary functions on the bottom:



Note that the bivariate summary functions count type 2 points (green) around type 1 points (red). Thus, points of pattern 1 are the focal points.

To view the different summary functions select the respective radio button and then the small “**ok**” button.

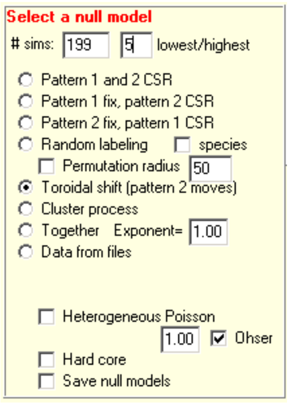
7. The next step is to select a **null model or point process** model implemented in *Programita*. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.



What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

□ Combine replicates
☑ Calculate simulation envelopes

A window will open that allows you to select a null model. In the example, we select “**Toroidal shift**”. In this case pattern 2 is shifted as a whole a random vector and points falling outside the observation window are wrapped following torus geometry.



Select a null model

# sims: 199 5 lowest/highest

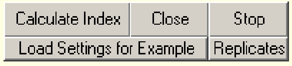
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling ☐ species
☐ Permutation radius 50
● Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

☐ Heterogeneous Poisson
1.00 ☑ Ohser
☐ Hard core
☐ Save null models

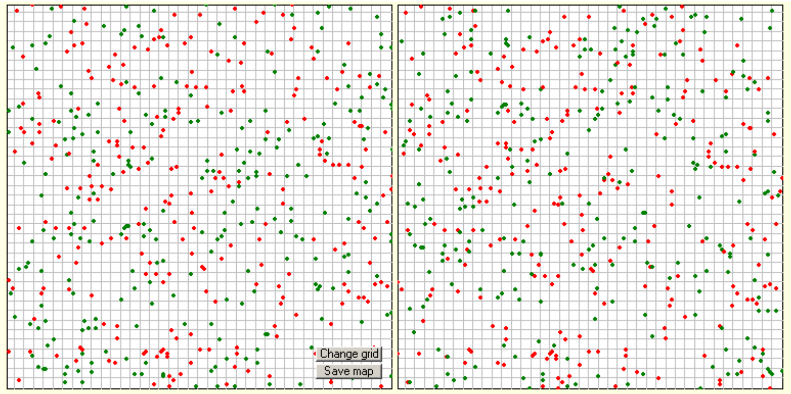
Here you can specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).

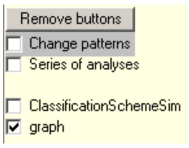
The checkbox "**Save null models**" allows you to save the patterns generated by the null model as "name_n.dat"

If all settings are specified; press "**Calculate Index**" and *Programita* conducts the simulations of the null model.



8. *Programita* shows the original point pattern being analyzed (left or top plot), and patterns of the Monte Carlo simulations of the null model (on the right or bottom) used for constructing the simulation envelopes and the GoF test.



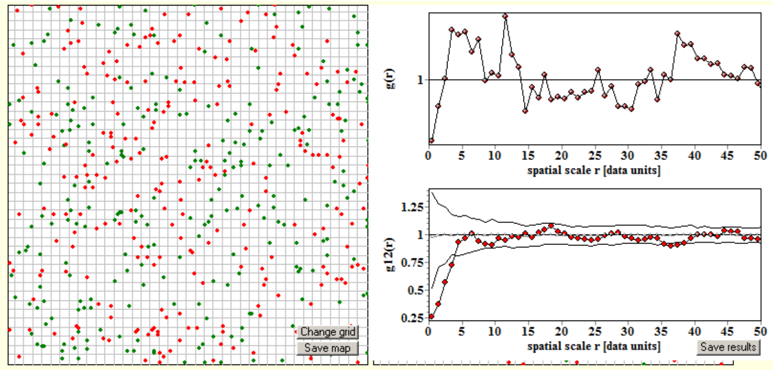


☐ Change patterns
☐ Series of analyses

☐ ClassificationSchemeSim
☑ graph

The simulation is quicker if *Programita* does not show the plots of all simulated data. You can not show the graphs by disabling the checkbox "**graph**" at the bottom right.

9. After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears:



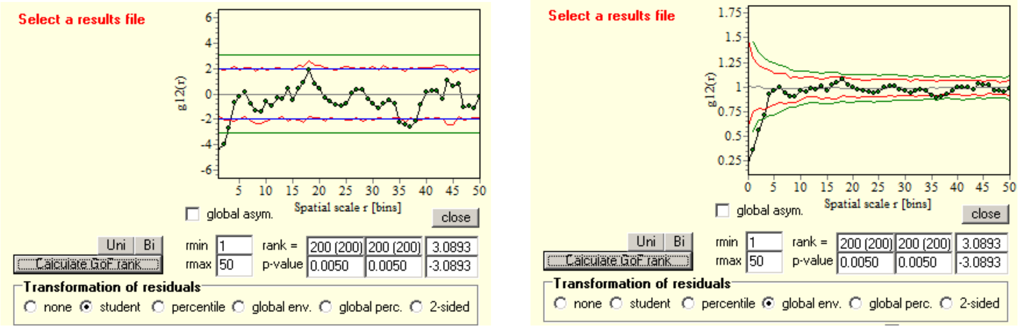
10. The top (or left) figure shows generally the results of the univariate analysis and the bottom (right) figure shows the results of the bivariate analysis. The null model did only displace the type 2 points, therefore no simulation envelopes appear for the univariate results.

11. To save **the results of the analysis for a particular summary functions** press the button Save results in the result graph for the bivariate analysis. *Programita* then generates a *.res file [e.g., "**name.res**" where "name" is a name] with the summary of the results and the settings of the analysis, and a *.env file with the detailed results of the summary function for the data and the simulations of the null model (the *.env file is named for example for the bivariate pair correlation function "g12(r)_name.env"). The *.env file can be used for the GoF test.

12. To conduct the GoF and global envelope tests check the small checkbox "**GoF**" that appears top right on the window "Select a null model". After enabling the check box a window appears where you need to click "**Uni**" or "**Bi**", depending if the analysis of interest is uni- or bivariate, respectively. Select "**Bi**" since the analysis was bivariate.

    A small graph with the observed summary function and the lowest and highest values of the null model appears. Provide now the distance interval (rmin, rmax) to be tested and click "**Calculate GoF rank**" for the GoF test based (Loosemore and Ford 2006) and the global envelope tests based on Myllymäki M., et al. (2015b).

Global envelope test of student transformed $g_{12}(r)$: Global envelopes that are variable in r



Select a results file

rmin 1 rank = 200 (200) 200 (200) 3.0893
rmax 50 p-value 0.0050 0.0050 -3.0893

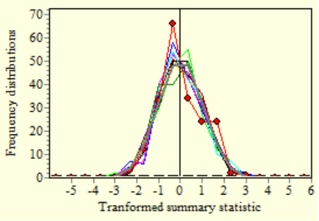
Transformation of residuals
○ none ● student ○ percentile ○ global env. ○ global perc. ○ 2-sided

Select a results file

rmin 1 rank = 200 (200) 200 (200) 3.0893
rmax 50 p-value 0.0050 0.0050 -3.0893

Transformation of residuals
○ none ○ student ○ percentile ● global env. ○ global perc. ○ 2-sided

The graph on the left shows the pointwise (red) and global (green) envelopes of the studentized summary function $S_i^{ses}(r)$ for different distances r. This transformation makes the pointwise simulation envelopes horizontal lines because all distributions of the $S_i^{ses}(r)$ follow the same standard normal distribution. The graph on the right shows the re-transformed global envelopes that are variable in r and which indicate a departure from the null model with significance level α if the observed summary function wanders at one or distances r outside the global envelopes (red).



Toroidal shift. Example *Book_Fig4_21a.res*

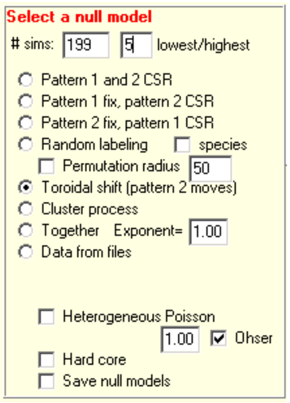
4.2 Methods for bivariate standard analysis

The following examples present step-by-step instructions for the most important bivariate analyses. We start with null models for independence. Devising a null model for independence for bivariate patterns is a highly non-trivial issue because a test of independence must maintain the univariate spatial structures in the two component patterns. *Programita* contains several bivariate null models that can be used as approximation of the independence null model.

4.2.1 Toroidal shift

Section 4.2.1 in Wiegand and Moloney (2014) deals with different approaches to test for independence between the two component patterns of a bivariate pattern. An early non-parametric solution is the toroidal shift null model where pattern 2 is shifted as a whole a random vector against pattern 1 which is fixed (see section 4.2.1.1 “The Toroidal Shift Null Model” in Wiegand and Moloney 2014). The parts of the pattern that are shifted outside the observation window re-appear following torus geometry. If one pattern is antecedent (e.g., the pattern of saplings relative to adult trees) the antecedent pattern (e.g., adults) should be selected as the fixed pattern 1 and the other randomized. If no pattern is antecedent, two tests should be conducted switching the role of type 1 and 2.

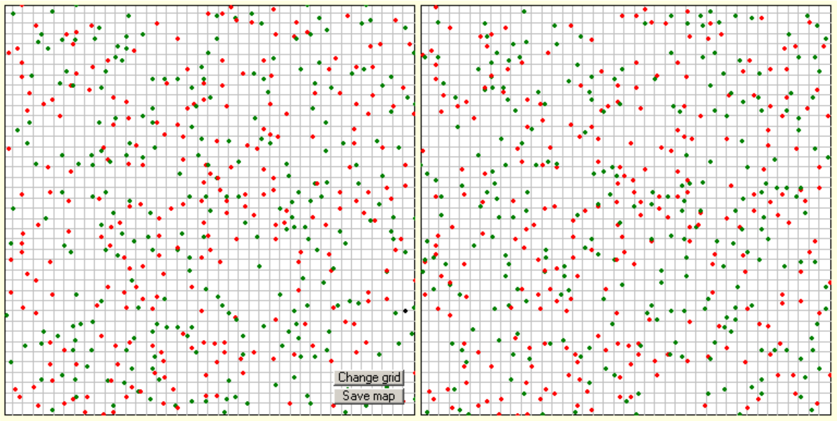
1. Execute *Programita*.
2. Highlight data file Book_Fig4_21a.dat you want to analyze in Input data and click small ok button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button “**Calculate Index**”.
7. Click the checkbox “**Calculate simulation envelopes**” in the menu “What do you want to do?” on the top left of the interface.
8. Specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).
9. Select “**Toroidal shift**”.
10. Press button “**Calculate Index**” and *Programita* shows the observed pattern (left) and the null model pattern (right):


Select a null model

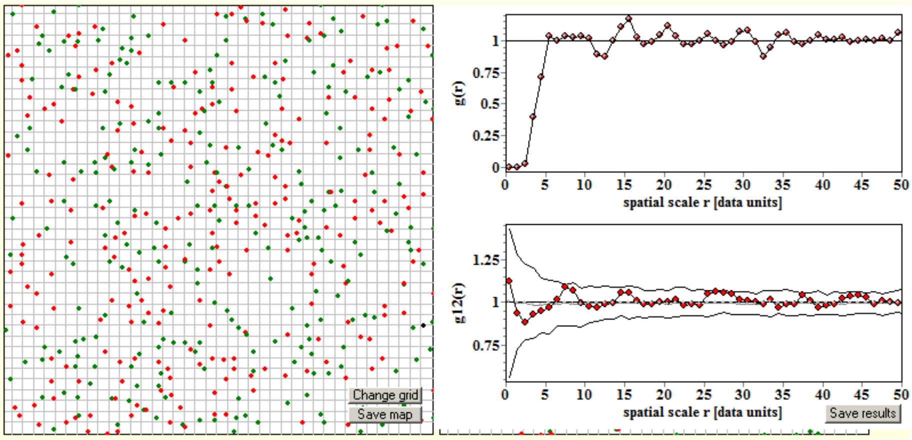
# sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
● Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

□ Heterogeneous Poisson
1.00 ☑ Ohser
□ Hard core
□ Save null models



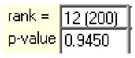
13. After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears:



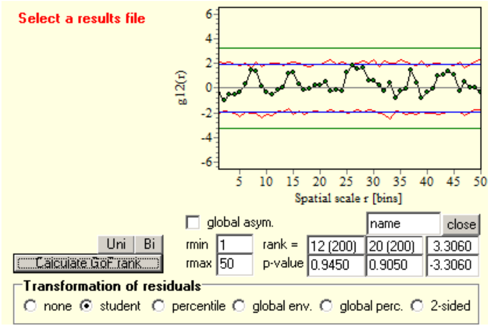
14.

The pair correlation function is fully within the pointwise simulation envelopes. This is also confirmed by the GoF tests over the distance interval 1 - 50m:

GoF test based on test statistic u_i of Loosmore and Ford and the global test envelope test based on the studentised summary statistic (left) and the global envelope test (right):



rank = 12 (200)
p-value 0.9450



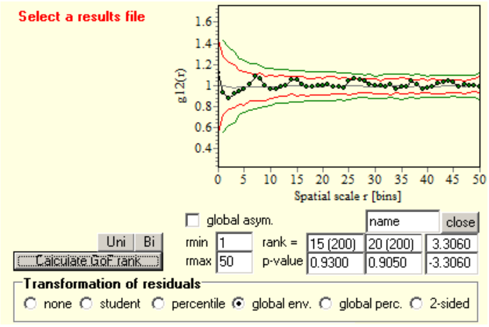
Select a results file

□ global asym. name

rmin 1
rmax 50

rank = 12 (200) 20 (200) 3.3060
p-value 0.9450 0.9050 -3.3060

Transformation of residuals
○ none ● student ○ percentile ○ global env. ○ global perc. ○ 2-sided



Select a results file

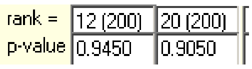
□ global asym. name

rmin 1
rmax 50

rank = 15 (200) 20 (200) 3.3060
p-value 0.9300 0.9050 -3.3060

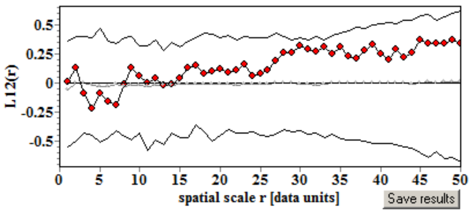
Transformation of residuals
○ none ○ student ○ percentile ● global env. ○ global perc. ○ 2-sided

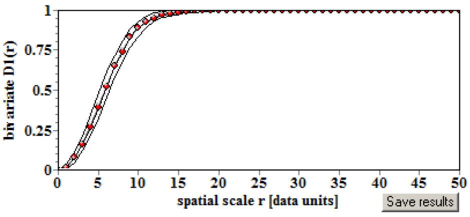
The rank and p-value of the global tests are given right hand of that of that of the test statistic u_i



rank = 12 (200) 20 (200)
p-value 0.9450 0.9050

15. The other summary functions are also well inside the pointwise simulation envelopes:



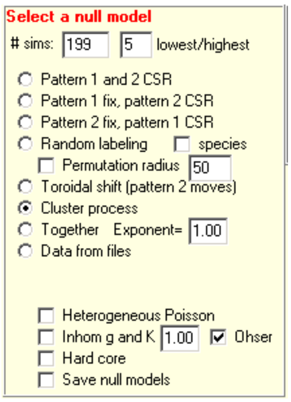


Parametric null model for independence. Example *Book_Fig4_24.res*

4.2.2 Pattern 2 Thomas process

You can also fit a parametric point process to one of the component patterns and use the realizations of the fitted point process as null model patterns for independence (see section "4.2.1.2 Parametric Point-Process Models" in Wiegand and Moloney 2014). In this case you define this pattern as pattern 2, fix the other pattern, and randomize pattern 2 following this point process. For a simple Thomas process this procedure is directly implemented in *Programita*. Otherwise, conduct first a univariate analysis and save the patterns generated by the fitted univariate Thomas sprocess. They can then be used as null model patterns for pattern 2 using the "from file" option. Here we show the null model where pattern 1 is fixed and a Thomas process fitted to pattern 2.

1. Execute *Programita*.
2. Highlight data file Fig4_24.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**".
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?".
8. A window will open that allows you to select a null model. Here you can specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).
9. In the example, select "**Cluster process**" and then select "**Pattern 1 fixed, patter 2 cluster**" in the window "Null models":

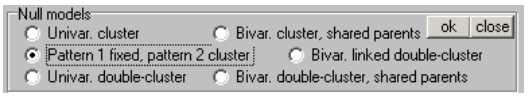


Select a null model

sims: 199 5 lowest/highest

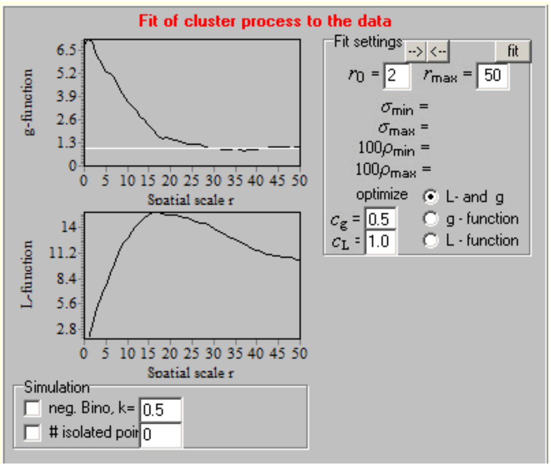
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
● Cluster process
○ Together Exponent= 1.00
○ Data from files

□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models



Null models
○ Univar. cluster ○ Bivar. cluster, shared parents
● Pattern 1 fixed, pattern 2 cluster ○ Bivar. linked double-cluster
○ Univar. double-cluster ○ Bivar. double-cluster, shared parents

10. Click the small "ok" button, the interface for fitting appears:

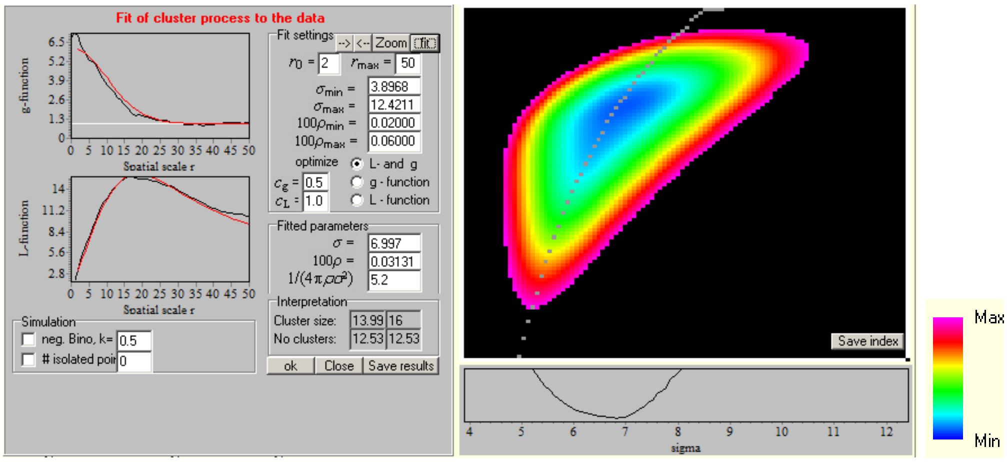


Fit of cluster process to the data

Fit settings
r_0 = 2 r_{max} = 50
σ_{min} =
σ_{max} =
$100\rho_{min}$ =
$100\rho_{max}$ =
optimize ● L- and g ○ g - function ○ L - function
c_g = 0.5
c_L = 1.0

Simulation
□ neg. Bino, k= 0.5
□ # isolated poi 0

11. Select the radio button “**L- and g**” to use the *g*- and *L*-function for fitting. The default settings over distance interval 2 to 50m (r_0 = 2 r_{max} = 50). (There are usually very little point pairs at distance r = 1 which results in high uncertainty, therefore better start with r = 2.
12. Click the button “**fit**” and *Programita* fits the two parameters ρ and σ of the Thomas process to the pattern. Note that ρA yields the number of clusters and 2σ the approximate radius of the “typical cluster”. To iteratively encircle the parameter space around the minimum in the σ-ρ parameter space click “**Zoom**” and “**Fit**”:



Fit of cluster process to the data

Fit settings

r_0 = 2 r_{max} = 50

σ_{min} = 3.8968

σ_{max} = 12.4211

$100\rho_{min}$ = 0.02000

$100\rho_{max}$ = 0.06000

optimize ● L - and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 6.997

100ρ = 0.03131

$1/(4\pi\rho\sigma^2)$ 5.2

Interpretation

Cluster size: 13.99 16

No clusters: 12.53 12.53

Simulation

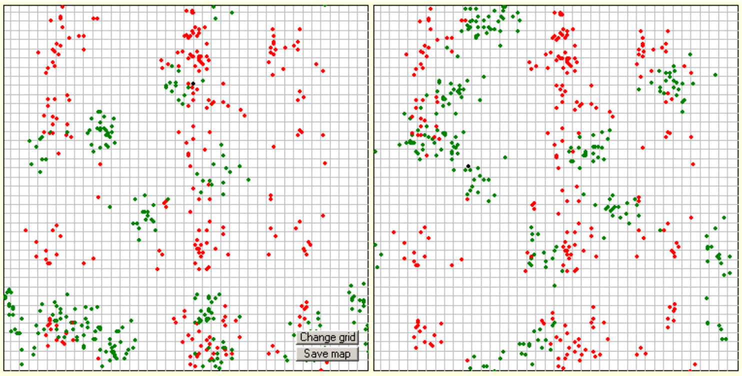
□ neg. Bino, k= 0.5

□ # isolated poi 0

The graph on the right shows the deviation between observed summary function (here the pair correlation and the *L*-function) and that predicted by Thomas process over the σ-ρ parameter space indicated by σ_{min}, σ_{max}, $100\rho_{min}$, and $100\rho_{min}$. There is a clear minimum at σ = 7.0 and ρA = 12.5 clusters. The pattern was generated with σ = 6.2 and ρA = 12.5 clusters.

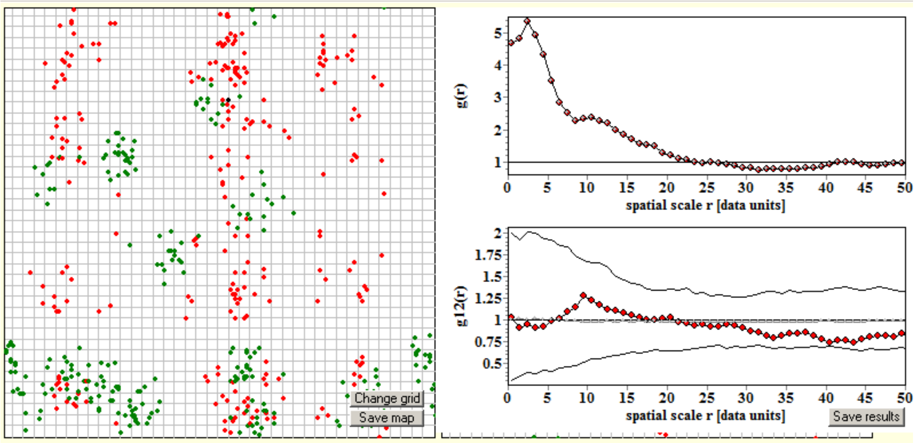
If you are satisfied with the fit, press the small “**ok**” button in the “Fitted parameter” section of the fitting window.

13. Press button “**Calculate Index**” and *Programita* shows the observed pattern (left) and the null model pattern (right):



Parametric null model for independence. Example *Book_Fig4_24.res*

16. After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears:



As expected, the bivariate pair correlation function and the other summary functions are fully within the pointwise simulation envelopes.

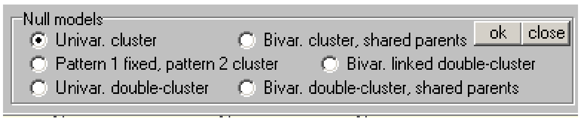
4.2.3 Pattern 2 from file

In this example we use null model patterns for pattern 2 that were generated previously with *Programita* fitting a Thomas process to the data of pattern 1. Thus, first fit a Thomas process to pattern 2 and save the patterns generated by the Thomas process. Second, read these files into *Programita* as null model for pattern 2.

1) Fit a Thomas process to pattern 2 (data Book_Fig4_24_p2.dat).

1. Execute *Programita*.
2. Highlight data file Book_Fig4_24_p2.dat you want to analyze in Input data.
3. Select "**no grid**" in What do you want to do?
4. Select bin of 1m window Select a new cell size
5. Select a ring width of 3 in the menu "Which method will you use"
6. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
7. Press button "**Calculate Index**".

14. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
15. A window will open that allows you to select a null model. Here you can specify the **number of simulations of the null model** (19 in the example).
16. In the example, select "**Cluster process**" and then select "**Univar. cluster**" in the window "Null models":



Null models

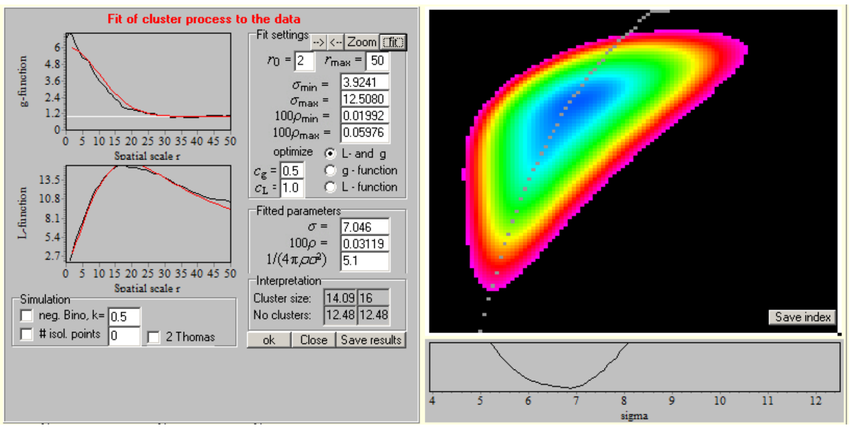
● Univar. cluster ○ Bivar. cluster, shared parents ok close

○ Pattern 1 fixed, pattern 2 cluster ○ Bivar. linked double-cluster

○ Univar. double-cluster ○ Bivar. double-cluster, shared parents

8. Now the interface for fitting appears.

To iteratively encircle the parameter space around the minimum in the σ-ρ parameter space click “**Zoom**” and “**Fit**”. As expected, the best fit is similar to that in the previous example:



Fit of cluster process to the data

Fit settings

r_0 = 2 r_{max} = 50

σ_{min} = 3.9241

σ_{max} = 12.5080

$100\rho_{min}$ = 0.01992

$100\rho_{max}$ = 0.05976

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 7.046

100ρ = 0.03119

$1/(4\pi\rho\sigma^2)$ 5.1

Interpretation

Cluster size: 14.09 16

No clusters: 12.48 12.48

Simulation

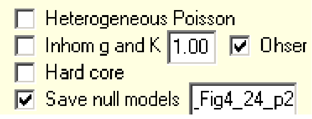
□ neg. Bino, k= 0.5

□ # isol. points 0 □ 2 Thomas

There is a clear minimum at σ = 7.0 and ρA = 12.5 clusters. The pattern was generated with σ = 6.2 and ρA = 12.5 clusters.

If you are satisfied with the fit, press the small “**ok**” button in the “Fitted parameter” section of the fitting window.

9. To save the patterns generated by the Thomas process click “Save null model” and provide name of null model files (Tho_Book_Fig4_24_p2)



□ Heterogeneous Poisson

□ Inhom g and K 1.00 ☑ Obser

□ Hard core

☑ Save null models _Fig4_24_p2

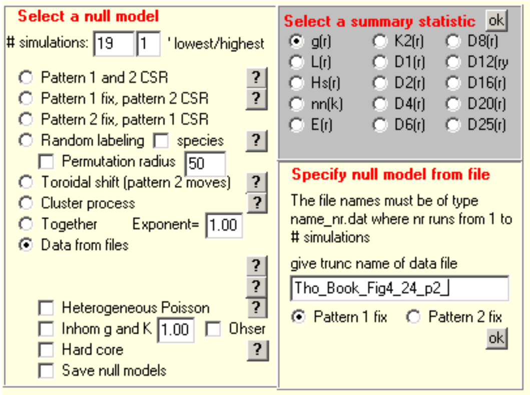
10. Press button “**Calculate Index**” and *Programita* generates the null model patterns Tho_Book_Fig4_24_p2_1.dat, Tho_Book_Fig4_24_p2_2_dat,….

2) Conduct analysis with the null model from file (Book_Fig4_24file.res)

11. Execute *Programita*.
12. Highlight data file Book_Fig4_24.dat you want to analyze in Input data and click the small “ok” button
13. Select bin of 1m window Select a new cell size
14. Select a ring width of 3 in the menu “Which method will you use”
15. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
16. Press button “**Calculate Index**”
17. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.

Parametric null model for independence. Example *Book_Fig4_24file.res*

18. Select “**Data from file**” in the window “Select a null model”.



Select a null model
# simulations: 19 1 'lowest/highest
○ Pattern 1 and 2 CSR ?
○ Pattern 1 fix, pattern 2 CSR ?
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species ?
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves) ?
○ Cluster process ?
○ Together Exponent= 1.00
● Data from files
?
?
□ Heterogeneous Poisson ?
□ Inhom g and K 1.00 □ Ohser
□ Hard core ?
□ Save null models

Select a summary statistic ok
● g(r) ○ K2(r) ○ D8(r)
○ L(r) ○ D1(r) ○ D12(ry
○ Hs(r) ○ D2(r) ○ D16(r)
○ nn(k) ○ D4(r) ○ D20(r)
○ E(r) ○ D6(r) ○ D25(r)

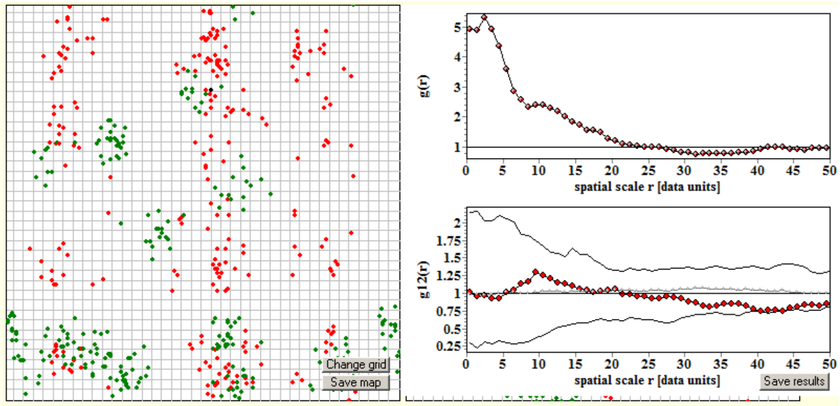
Specify null model from file
The file names must be of type name_nr.dat where nr runs from 1 to # simulations
give trunc name of data file
Tho_Book_Fig4_24_p2_
● Pattern 1 fix ○ Pattern 2 fix
ok

19. Insert the name trunk of the null model files (Tho_Book_Fig4_24_p2_) in the window “Specify null model files from file” that opens to read the null model files Tho_Book_Fig4_24_p2_1.dat, Tho_Book_Fig4_24_p2_2.dat, …

    Click also the radio button “**Pattern 1 fix**”. This means that the null model files are used for pattern 2.

    To finish click the small **ok** button in the window “Specify null model files from file”. Specify the number of simulations of the null model (19 in the example) and the rule for the estimation of simulation envelopes (here the 1th lowest and highest values of the summary function of the 19 simulated null model data sets).

20. If all settings are specified, press the button “**Calculate Index**” and *Programita* conducts the simulations of the null model.
21. After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears:

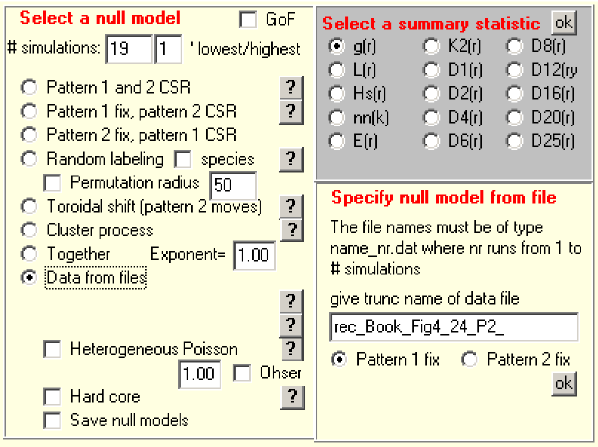


As in the previous example, the bivariate pair correlation function and the other summary functions are fully within the pointwise simulation envelopes.

4.2.4 Pattern reconstruction

In this example we use null model patterns for pattern 2 that were generated previously with the pattern reconstruction software (Wiegand et al. 2013). *Programita* can read these files and use it for the null model.

1. Execute *Programita*.
2. Highlight data file Book_Fig4_24.dat you want to analyze in Input data
3. Select "**no grid**" in What do you want to do?
4. Select bin of 1m window Select a new cell size
5. Select a ring width of 3 in the menu "Which method will you use"
6. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
7. Press button "**Calculate Index**"
8. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
9. Select "**Data from file**" in the window "Select a null model".



Select a null model □ GoF
# simulations: 19 1 ' lowest/highest
○ Pattern 1 and 2 CSR ?
○ Pattern 1 fix, pattern 2 CSR ?
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species ?
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves) ?
○ Cluster process ?
○ Together Exponent= 1.00
● Data from files
?
?
□ Heterogeneous Poisson ?
1.00 □ Ohser
□ Hard core ?
□ Save null models

Select a summary statistic ok
● g(r) ○ K2(r) ○ D8(r)
○ L(r) ○ D1(r) ○ D12(r)
○ Hs(r) ○ D2(r) ○ D16(r)
○ nn(k) ○ D4(r) ○ D20(r)
○ E(r) ○ D6(r) ○ D25(r)

Specify null model from file
The file names must be of type name_nr.dat where nr runs from 1 to # simulations
give trunc name of data file
rec_Book_Fig4_24_P2_
● Pattern 1 fix ○ Pattern 2 fix
ok

10. Insert the name trunk of the null model files (rec_Book_Fig4_24_p2_) in the window "Specify null model files from file" that opens to read the null model files rec_Book_Fig4_24_p2_1.dat, rec_Book_Fig4_24_p2_2.dat, …

    Click also the radio button "**Pattern 1 fix**". This means that the null model files are used for pattern 2.

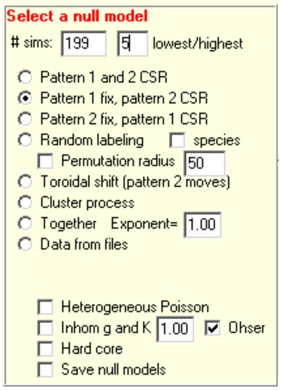
    To finish click the small **ok** button in the window "Specify null model files from file". Specify the number of simulations of the null model (19 in the example) and the rule for the estimation of simulation envelopes (here the 1th lowest and highest values of the summary function of the 19 simulated null model data sets).

11. If all settings are specified, press the button "**Calculate Index**" and *Programita* conducts the simulations of the null model.
12. After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears.

4.2.5 Pattern 2 CSR

In this example we use the CSR null model for pattern 2. Note that this is not a suitable null model for testing for independence if pattern 2 shows clustering or hyperdispersion.

1. Execute *Programita*.
2. Highlight data file Book_Fig4_24.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Pattern 1 fix, pattern 2 CSR**" in the window "Select a null model".
9. Specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).



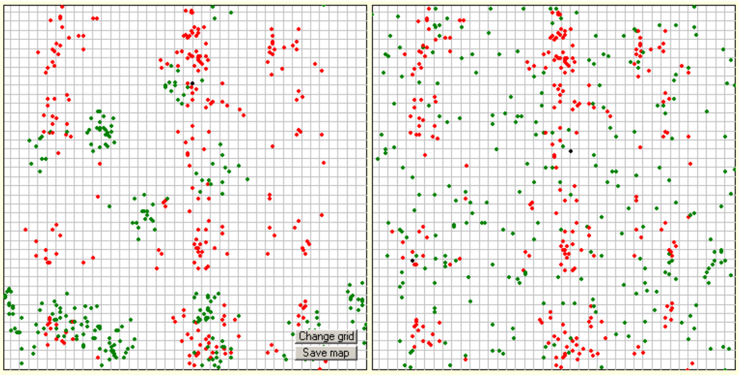
Select a null model

sims: 199 | 5 lowest/highest

○ Pattern 1 and 2 CSR
● Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

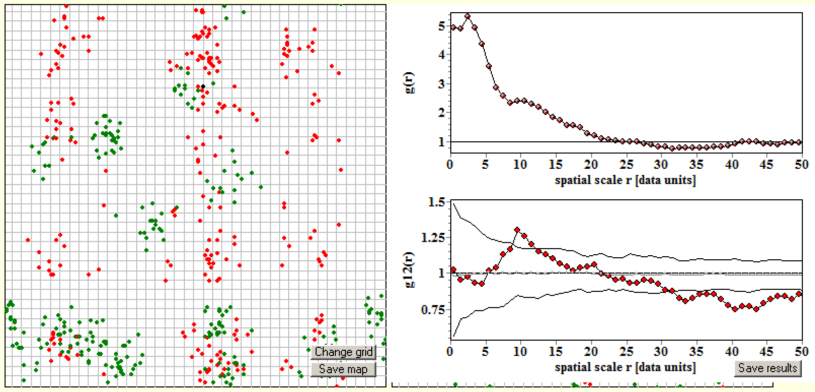
10. If all settings are specified, press the button "**Calculate Index**" and *Programita* conducts the simulations of the null model. The null model patterns are shown on the right, the observed pattern on the left:



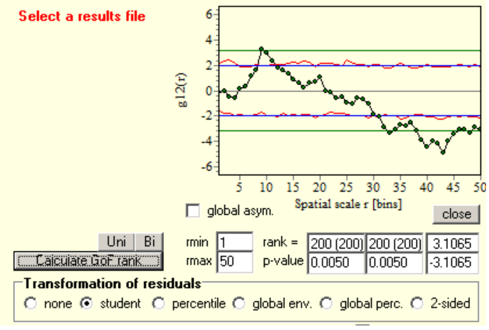
It is clear that the null model (i.e., the green points) does not conserve the observed cluster structure.

11. After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears. Note that no simulation envelopes are shown for the univariate case because pattern 1 is fixed.

    The simulation pointwise envelopes for the bivariate summary functions are substantially narrower than that of the diverse null models that conserved the observed spatial structure of pattern 2.



12. The GoF and global envelope tests show that the departures of the bivariate pair correlation function are significant:
13.



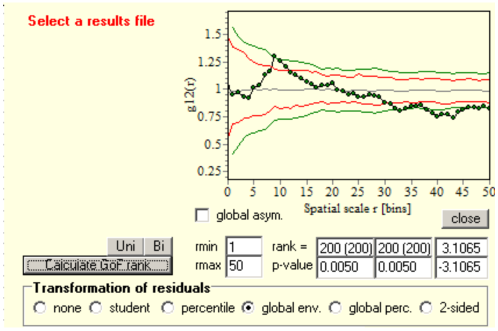
□ global asym.

rmin 1 rank = 200 (200) 200 (200) 3.1065

rmax 50 p-value 0.0050 0.0050 -3.1065

Transformation of residuals

○ none ● student ○ percentile ○ global env. ○ global perc. ○ 2-sided



□ global asym.

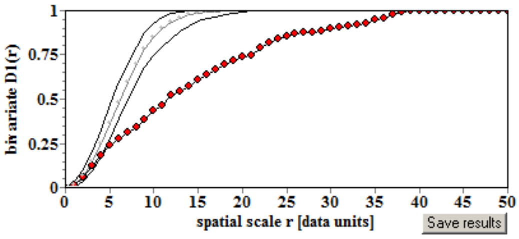
rmin 1 rank = 200 (200) 200 (200) 3.1065

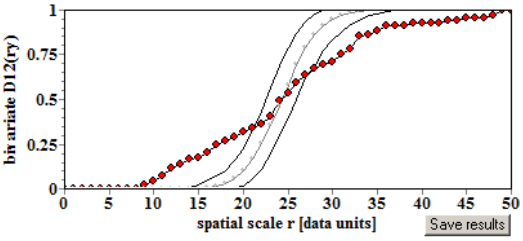
rmax 50 p-value 0.0050 0.0050 -3.1065

Transformation of residuals

○ none ○ student ○ percentile ● global env. ○ global perc. ○ 2-sided

14. Especially the nearest neighbor distribution functions show strong departures from the null model:





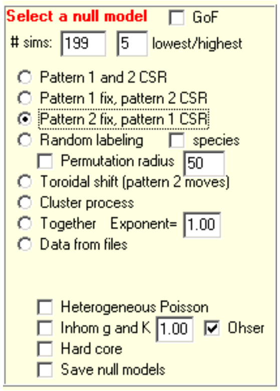
The nearest type 2 neighbor of type 1 points in the null model is usually much closer than in the observed data. This is because pattern 2 was clustered in the data.

The observed bivariate pattern contains non-random spatial structure, but this structure is caused by the univariate clustering of the component patterns which accidently created a point configuration which is difficult to reproduce with the null model where pattern 2 is CSR. If we conserve the observed univariate structures (as in example *Book_Fig4_24.res*) we see that such point configurations can arise just by chance.

4.2.6 Pattern 1 CSR

In this example we use the CSR null model for pattern 1 whereas pattern 2 is unchanged. Note that this is not a suitable null model for testing for independence if pattern 1 shows clustering or hyperdispersion.

1. Execute *Programita*.
2. Highlight data file Book_Fig4_24.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Pattern 2 fix, pattern 1 CSR**" in the window "Select a null model".
9. Specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).



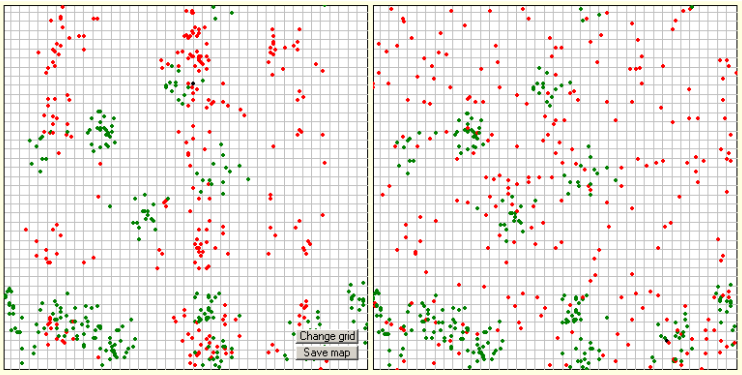
Select a null model □ GoF

sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
● Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

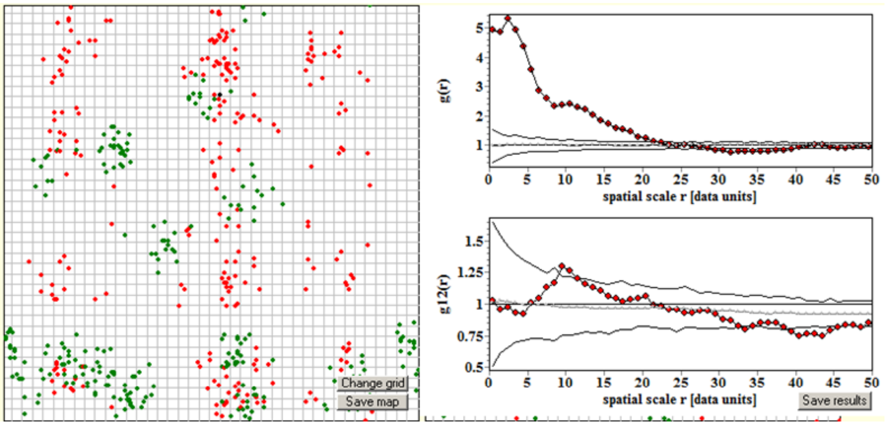
□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

10. If all settings are specified, press the button "**Calculate Index**" and *Programita* conducts the simulations of the null model. The null model patterns are shown on the right, the observed pattern on the left:

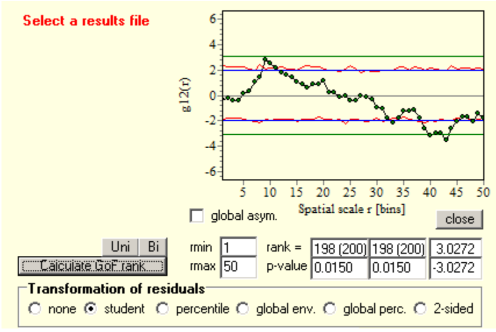


It is clear that the null model (i.e., the red points) does not conserve the observed cluster structure.

11. After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears. Note that there is now a result for the univariate analysis because pattern 1 was randomized.



12. The global simulation envelopes indicate a weak but significant effect but the standard GoF test of the untransformed data is not significant (right graph).



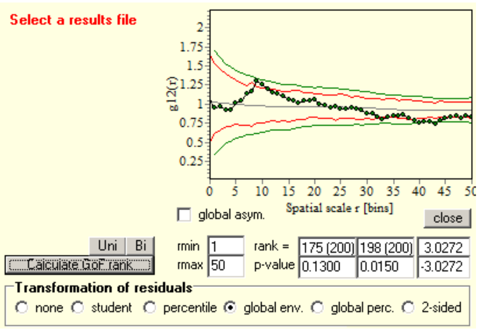
□ global asym.

rmin 1 rank = 198 (200) 198 (200) 3.0272

rmax 50 p-value 0.0150 0.0150 -3.0272

Transformation of residuals

○ none ● student ○ percentile ○ global env. ○ global perc. ○ 2-sided



□ global asym.

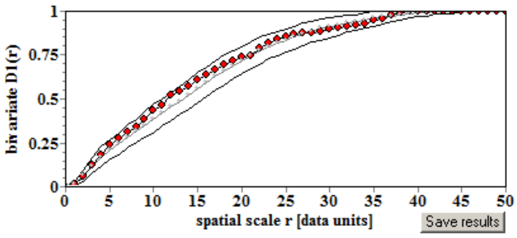
rmin 1 rank = 175 (200) 198 (200) 3.0272

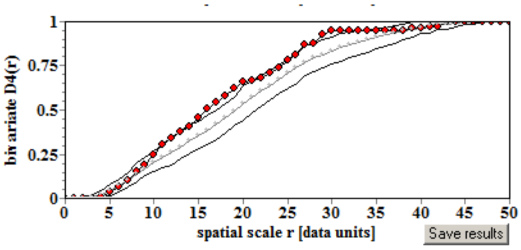
rmax 50 p-value 0.1300 0.0150 -3.0272

Transformation of residuals

○ none ○ student ○ percentile ● global env. ○ global perc. ○ 2-sided

13. Interestingly, the nearest neighbor distribution functions show no departures from the null model, but those with higher neighborhood ranks such as the 4th nearest neighbor:



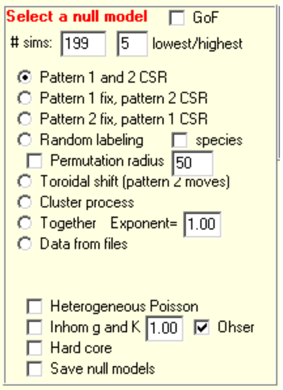


However, as expected there is a strong departure from the CSR null model in the univariate analysis.

4.2.7 Pattern 1 and 2 CSR

In this example we use the CSR null model for both component patterns. Note that this is not a suitable null model for testing for independence if pattern 1 and 2 shows clustering or hyperdispersion.

1. Execute *Programita*.
2. Highlight data file Book_Fig4_24.dat you want to analyze in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button “**Calculate Index**”
7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
8. Select “**Pattern 1 fix, pattern 2 CSR**” in the window “Select a null model”.
9. Specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).



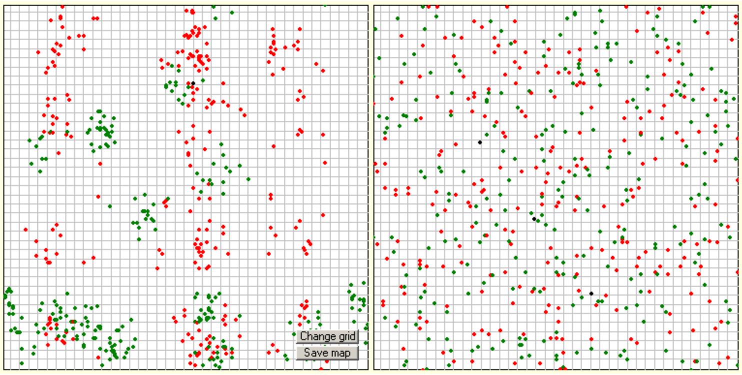
Select a null model □ GoF

# sims: 199 5 lowest/highest

● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

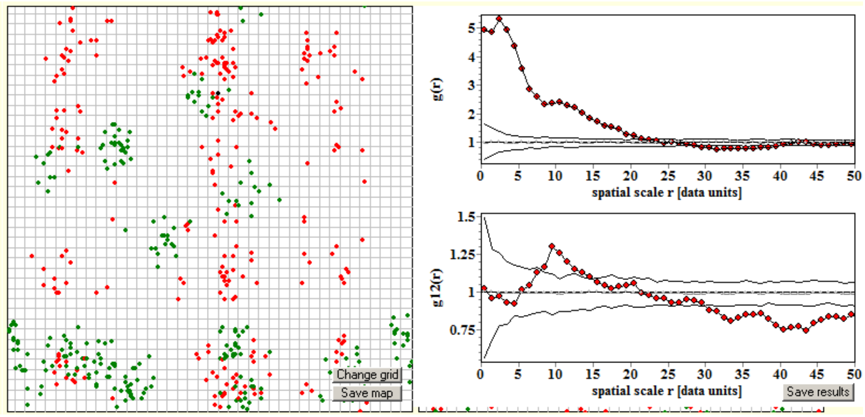
□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

10. If all settings are specified, press the button “**Calculate Index**” and *Programita* conducts the simulations of the null model. The null model patterns are shown on the right, the observed pattern on the left:

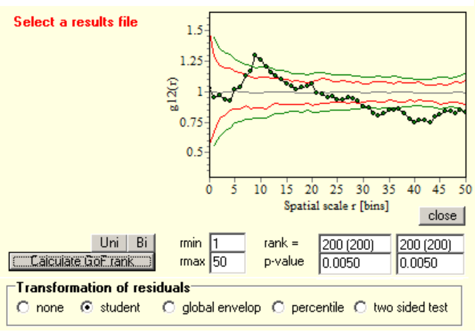


It is clear that the null model (i.e., the red and green points) does not conserve the observed cluster structure.

11. After the simulations of the null model the figure with the simulated patterns of the null model disappears, and a figure with the result of the analysis appears. The pointwise simulation envelopes are substantially narrower than for the diverse null models that conserved the observed spatial structure of pattern 2.

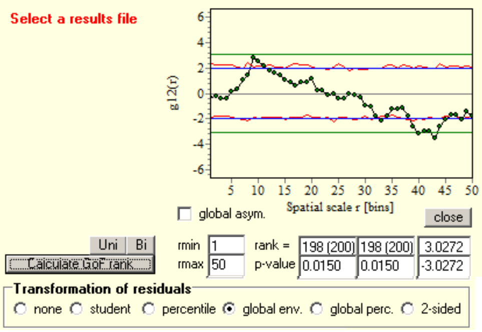


12. As a consequence, the GoF tests show that the departures of the bivariate pair correlation function are highly significant:



rmin 1
rmax 50
rank = 200 (200) 200 (200)
p-value 0.0050 0.0050

Transformation of residuals
○ none ● student ○ global envelop ○ percentile ○ two sided test

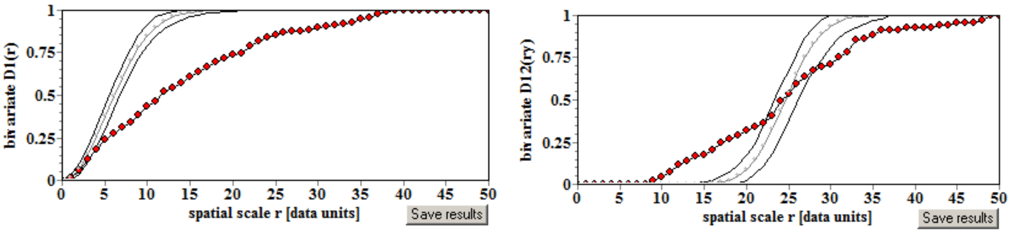


□ global asym.

rmin 1
rmax 50
rank = 198 (200) 198 (200) 3.0272
p-value 0.0150 0.0150 -3.0272

Transformation of residuals
○ none ○ student ○ percentile ● global env. ○ global perc. ○ 2-sided

Especially the nearest neighbor distribution functions show strong departures from the null model:



The nearest neighbor in the null model is usually much closer than in the observed data. This is because the observed patterns 1 and 2 were clustered.

The observed bivariate pattern contains non-random spatial structure, but this structure is caused by the univariate clustering of the component patterns which created a point configuration which is impossible to reproduce with the null model where patterns 1 and 2 are CSR. However, if we conserve the observed univariate structures we see that such point configurations can arise just by chance.

4.2.8 Classification scheme

The classification scheme is used for exploring the overall spatial association structure of species pairs in a multivariate pattern. The goal of this analysis is to determine how the individuals of a species j were distributed within local neighborhoods of individuals of a species i, and this analysis is repeated for all pairs i–j of the community. The scheme is basically a summary of all pairwise bivariate analyses at a given neighborhood scale *r* which are possible for a community. Sections 4.2.2.2- 4.2.2.4 and section 4.3.1.1 in Wiegand and Moloney (2014) and Getzin et al. (2014) provide details on the scheme. Applications of the scheme include Wiegand et al. (2007b), Martínez et al. (2010), Wang et al. (2010), Wiegand et al. (2012), and Jacquemyn et al. (2014).

The bivariate analysis of the scheme can use different null models; in the simplest case one pattern is kept fixed and the other pattern is randomized following CSR, or one pattern is kept fixed and the other pattern is randomized following a toroidal shift. Additionally, you can use null model patterns that are produced outside Programita, for example, by pattern reconstruction.

The analysis using the CSR null model reveals how frequently the two different species were in close contact and therefore have the opportunity to interact whereas the toroidal shift null (and the homogeneous pattern reconstruction) model explores in good approximation how frequent different departures from independence occur for the pairs of species at a given neighborhood *r*. Here departures from independence con occur due to both, species interactions (e.g., repulsion due to competition) or shared or opposed habitat associations. The inhomogeneous pattern reconstruction null model that keeps the larger-scale intensity of the observed pattern and randomizes only the smaller-scale spatial structures of the univariate patterns. Here departures from independence con occur in good approximation only due to species interactions (e.g., repulsion due to competition).

The scheme relies on two summary functions, the bivariate *K* function $K_{ij}(r)$ and the bivariate distribution function $D_{ij}(r)$ of the distances to the nearest neighbor and their expectations under the two null models. Why do we need two summary functions? In case of homogeneous patterns departures in $D_{ij}(r)$ and $K_{ij}(r)$ would be correlated. However, for “real world” heterogeneous patterns the local spatial configuration of individuals of species *j* around individuals of species i can widely vary at different locations in the plot. For example, at some locations some individuals of species *i* may have many neighbors of species *j* and at other locations some individuals of species *i* may have only few neighbors of species *j*. To describe the different types of spatial configurations which may arise we classify the bivariate pattern of a species pair at neighborhoods *r* into a two-dimensional space spanned by the two axis:

$$\hat{P}(r) \quad = \quad \hat{D}_{ij}(r) - D_{ij}^{null}(r)$$

$$\hat{M}(r) \quad = \quad \ln(\hat{K}_{ij}(r)) - \ln(K_{ij}^{null})$$

where the hat symbol indicates the observed value of each species pair. The $D_{ij}(r)$ and $K_{ij}(r)$ axes were transformed in a way that the expectation of independence or CSR) is located at the origin and that positive or negative departures of $K(r)$ from the null model are weighted in the same way (Wiegand et al. 2012).

An alternative definition of the axes of the scheme that correct for the variability in the null model (based on standardized effect sizes) is given by (Getzin et al. 2014):

$$\hat{P}^{eff}(r) \quad = \quad [\hat{D}_{ij}(r) - D_{ij}^{null}(r)] / SD[D_{ij}^{null}(r)]$$

$$\hat{M}^{eff}(r) \quad = \quad [\hat{K}_{ij}(r) - K_{ij}^{null}(r)] / SD[K_{ij}^{null}(r)]$$

where *SD*[] estimates the standard deviation over all simulations of the null model. Because the distribution of *P*(*r*) and *M*(*r*) under the null model can be approximated by the standard normal distribution, the box delimited by values of –2.33, 2.33 (which correspond to a p-value of 0.025 for two summary statistics individually) approximates the area where the null hypothesis cannot be rejected, and a given species departs more strongly from independence the farther away it is located from the box.

However, the quadrant of the scheme where the species is located provides additional information on the type of departure. Four fundamental types of spatial association patterns are possible for each neighborhood *r* [32]:

- **Type 0: no departures:** neither $K_{12}(r)$ nor $D_{12}(r)$ show significant departure from the null model
- **Type I: Segregation:** Species pairs located in the lower-left quadrant show segregation because there are fewer individuals of species *j* within neighborhoods of radius *r* around individuals of species *i* than expected under the null model [$\hat{P}(r) < 0$ and $\hat{M}(r) < 0$].
- **Type II: Partial overlap:** Species pairs located in the upper-left quadrants show partial overlap because individuals of species *j* occur more often within neighborhoods of radius *r* around individuals of species *i* [$M(r) > 0$], but a notable proportion of individuals of species *i* have fewer neighbors of species *j* [$P(r) < 0$] than expected under the null model.
- **Type III: Mixing:** Species pairs located in the upper-right quadrant show a high degree of spatial association (mixing) because here individuals of species *j* occur more often within neighborhoods of radius *r* around individuals of species *i* [$M(r) > 0$], and individuals of species *i* have more neighbors of species *j* [$P(r) > 0$], than expected under the null model.
- **Type IV:** For species pairs located in the lower-right quadrant, species *i* individuals are highly clustered and some species *j* individuals occur in these clusters [$\hat{P}(r) > 0$ and $\hat{M}(r) < 0$]. This type rarely occurs.

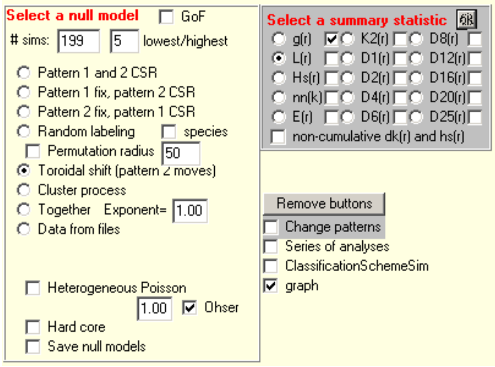
Instructions for the scheme based on univariate pattern files

1. Prepare a data set with a univariate *.dat data file for each species. No specific name conventions are required, however, it is better to code the name with the species acronyms (e.g., ADE1TR.dat, ALSEBL.dat, BEILPE.dat) or numbers (e.g., BCI_C1_sp1.dat, BCI_C1_sp2.dat, BCI_C1_sp3.dat,…). In the example data we have Competition1_sp1.dat, Competition1_sp2.dat, ... The latter data files correspond to a simulated community (of Miller et al. 2017) with small-scale competition. Do not include species with less than 50 individuals.

2. Prepare one bivariate data file. In the example this is Competition1_sp1_sp2.
3. Prepare an *.txt ASCII file with a list of the species names (without extension) you want to analyze. In our case the list is "Competition1.txt":

```
Competition1_sp1
Competition1_sp2
Competition1_sp3
....
```

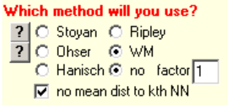
4. Now prepare the example analysis for the bivariate data file. First, execute *Programita*.
5. Highlight data file Competition1_sp1_sp2.dat you want to analyze in Input data and click the small "ok" button.
1. Select bin of 1m window Select a new cell size
2. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
3. Select no edge correction for $D_{12}(r)$ because Hanisch edge correction does not work well for some bivariate patterns (see section 3.1.4.5 in Wiegand and Moloney 2014)
4. Press button "**Calculate Index**"
5. Select L(r) in window "Select a summary function" and click the small "**ok**" button
6. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface. Select "**Toroidal shift**" in the window "Select a null model".
7. Specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).



Select a null model □ GoF
# sims: 199 5 lowest/highest
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
● Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files
□ Heterogeneous Poisson
1.00 ☑ Ohser
□ Hard core
□ Save null models

Select a summary statistic ok
○ g(r) ☑ ○ K2(r) □ ○ D8(r) □
● L(r) □ ○ D1(r) □ ○ D12(r) □
○ Hs(r) □ ○ D2(r) □ ○ D16(r) □
○ nn(k) □ ○ D4(r) □ ○ D20(r) □
○ E(r) □ ○ D6(r) □ ○ D25(r) □
□ non-cumulative dk(r) and hs(r)

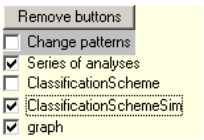
Remove buttons
□ Change patterns
□ Series of analyses
□ ClassificationSchemeSim
☑ graph



Which method will you use?
? ○ Stoyan ○ Ripley
? ○ Ohser ● WM
○ Hanisch ● no factor 1
☑ no mean dist to kth NN

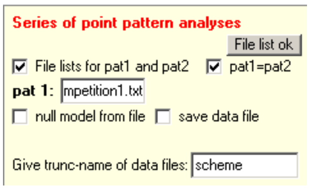
If all settings are specified, press the button "**Calculate Index**" and *Programita* conducts the simulations of the null model. Save the results with the "**Save results**" button (Scheme_competition1_sp1_2_torus.res) .

8. Enable check box "**ClassificationSchemeSim**" on the right hand side of the "Select a null model" window. A window opens with settings of the Series of analyses.



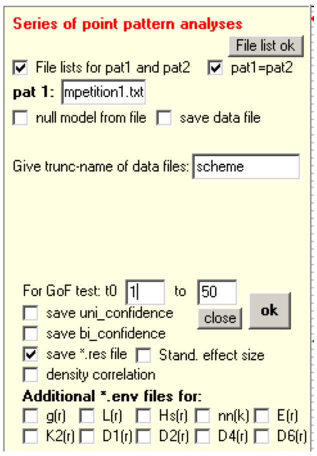
Remove buttons
□ Change patterns
☑ Series of analyses
□ ClassificationScheme
☑ ClassificationSchemeSim
☑ graph

9. To select files for pairwise analyses based on the univariate data files select "**File list for pat1 and pat2**".
10. If pattern 1 and 2 should be selected from the same list select "**Pat1=pat2**" (click the check box "**pat1=pat2**" two times). This is necessary to omit that the same file is selected as pattern 1 and pattern 2.
11. To save the bivariate data files that are automatically assembled by *Programita* enable "**save data file**"



Series of point pattern analyses
File list ok
☑ File lists for pat1 and pat2 ☑ pat1=pat2
pat 1: mpetition1.txt
□ null model from file □ save data file
Give trunc-name of data files: scheme

12. Insert the name of the file list for **pat 1** (and if appropriate, for pat 2). In our case it is “Competition1.txt”.
13. The trunk-name is used to name the summary output file. We use “scheme”
14. Provide the distance interval for the GoF test (1-50m in our case)
15. To obtain the second version of the scheme with standardized effect sizes click the check box “Stand. effect size”
16. Once all settings are specified, click the fat **ok** bottom ok, and then “**Calculate Index**” to start the series of analyses.
17. Disable the checkbox “graph” to simulate quicker



Series of point pattern analyses

File list ok

☑ File lists for pat1 and pat2 ☑ pat1=pat2

pat 1: mpetition1.txt

☐ null model from file ☐ save data file

Give trunc-name of data files: scheme

For GoF test: t0 1 to 50

☐ save uni_confidence

☐ save bi_confidence

☑ save *.res file ☐ Stand. effect size

☐ density correlation

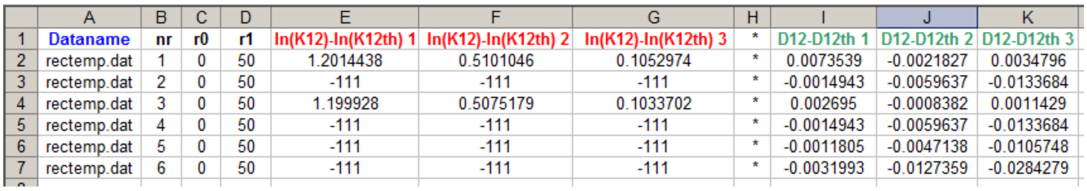
close ok

Additional *.env files for:

☐ g(r) ☐ L(r) ☐ Hs(r) ☐ nn(k) ☐ E(r)

☐ K2(r) ☐ D1(r) ☐ D2(r) ☐ D4(r) ☐ D6(r)

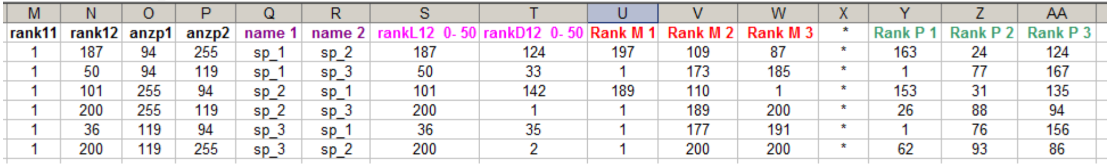
18. The important output file is then named “Summary_SchemeSim_scheme.txt” (“scheme” is the name you selected). It is a comma delimited ASCII file. It gives you a summary of the result of all analyses and can be used to construct the scheme. The first part of the data file:



	A	B	C	D	E	F	G	H	I	J	K
1	Dataname	nr	r0	r1	ln(K12)-ln(K12th) 1	ln(K12)-ln(K12th) 2	ln(K12)-ln(K12th) 3	*	D12-D12th 1	D12-D12th 2	D12-D12th 3
2	rectemp.dat	1	0	50	1.2014438	0.5101046	0.1052974	*	0.0073539	-0.0021827	0.0034796
3	rectemp.dat	2	0	50	-111	-111	-111	*	-0.0014943	-0.0059637	-0.0133684
4	rectemp.dat	3	0	50	1.199928	0.5075179	0.1033702	*	0.002695	-0.0008382	0.0011429
5	rectemp.dat	4	0	50	-111	-111	-111	*	-0.0014943	-0.0059637	-0.0133684
6	rectemp.dat	5	0	50	-111	-111	-111	*	-0.0011805	-0.0047138	-0.0105748
7	rectemp.dat	6	0	50	-111	-111	-111	*	-0.0031993	-0.0127359	-0.0284279

- Dataname: the datafile of the given species pair. It is always rectemp.dat if the bivariate data were composed from the univariate data. The columns “name 1” and “name 2” provided later give the names of the univariate files.
- nr: the number of the species pair analyzed
- r0 and r1: the interval of the GoF test
- the values of the M-axis of the scheme for distances 1, 2, 3 (ln(K12)-ln(K12th) 1, ln(K12)-ln(K12th) 2, ..., ln(K12)-ln(K12th) 50). We show here only distances *r* up to 3. A value of -111 indicates a large negative value (> -4)
- the values of the P-axis of the scheme for distances 1, 2, 3 (-D12-D12th 1, - D12-D12th 2, ..., - D12-D12th 50). We show here only distances *r* up to 3.

The second part of the file:



M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
rank11	rank12	anzp1	anzp2	name 1	name 2	rankL12 0- 50	rankD12 0- 50	Rank M 1	Rank M 2	Rank M 3	*	Rank P 1	Rank P 2	Rank P 3
1	187	94	255	sp_1	sp_2	187	124	197	109	87	*	163	24	124
1	50	94	119	sp_1	sp_3	50	33	1	173	185	*	1	77	167
1	101	255	94	sp_2	sp_1	101	142	189	110	1	*	153	31	135
1	200	255	119	sp_2	sp_3	200	1	1	189	200	*	26	88	94
1	36	119	94	sp_3	sp_1	36	35	1	177	191	*	1	76	156
1	200	119	255	sp_3	sp_2	200	2	1	200	200	*	62	93	86

- rank11: the rank of the GoF test for the univariate pair correlation function (not used here)
- rank12: the rank of the GoF test for the bivariate pair correlation function (additional info)
- anzp1: number of points of pattern 1
- anzp2: number of points of pattern 2
- name 1: name of focal pattern
- name 2: name of second pattern
- rankL12 0- 50: the rank of the GoF test for the $L_{ij}(r)$ over the entire range of distances selected (to check if the M-axis of scheme is significant)

- rankD12 0- 50: the rank of the GoF test for the $D_{ij}(r)$ over the entire range of distances 1-50 selected (to check if P-axis of the scheme is significant)
- Rank M1, Rank M2, …, Rank M 50: the rank of the M-axis of the scheme at all individual distances $r = 1, 2, \dots$ 50m analyzed
- Rank P1, Rank P 2, …, Rank P50: the rank of the P-axis of the scheme at all individual distances $r = 1, 2, \dots$ 50m analyzed

19. Based on the values of the *M* and *P* axes and the rank of the *M* and *P* axes at distance *r* you can determine the association type of a given species pair.

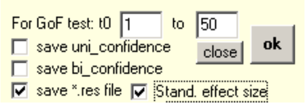
    Because we use two test statistics at the same time [$L_{ij}(r)$, $D_{ij}(r)$] we need to use a P-value of 0.025 for each summary functions to yield an overall error rate of 5%. If you selected 199 simulation of the null model, the rank must be therefore larger than 195 to be significant.

    If (Rank P r) > 195 or (Rank M r) > 195 the species pair belongs at distance *r* to a significant class:

- P < 0 and M < 0: type 1 (**segregation**)
- P < 0 and M > 0: type 2 (**partial overlap**)
- P > 0 and M > 0: type 3 (**mixing**)
- P > 0 and M < 0: type 4 (does only rarely occur)
- otherwise the species pair belongs to the no "**significant patterning class"**

20. **The scheme with effects sizes**

    If you selected "Stand. effect sizes"



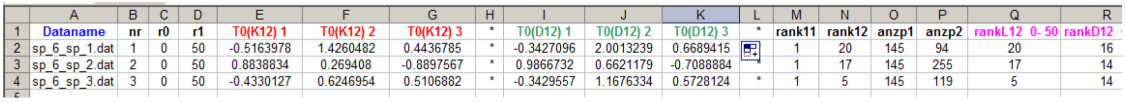
For GoF test: t0 1 to 50

□ save uni_confidence

□ save bi_confidence

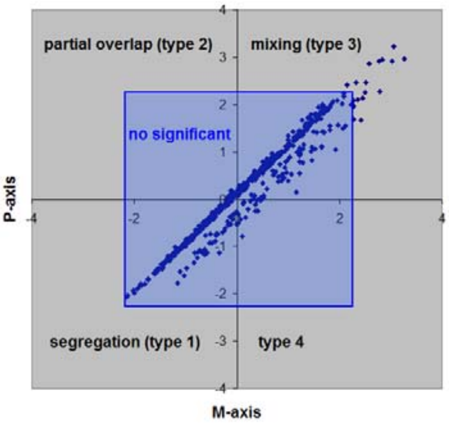
☑ save *.res file ☑ Stand. effect size

the corresponding summary file "Summary_SchemeSim_scheme.txt" is given by



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Dataname	nr	r0	r1	T0(K12) 1	T0(K12) 2	T0(K12) 3	*	T0(D12) 1	T0(D12) 2	T0(D12) 3	*	rank11	rank12	anzp1	anzp2	rankL12 0- 50	rankD12
2	sp_6_sp_1.dat	1	0	50	-0.5163978	1.4260482	0.4436785	*	-0.3427096	2.0013239	0.6689415		1	20	145	94	20	16
3	sp_6_sp_2.dat	2	0	50	0.8838834	0.269408	-0.8897567	*	0.9866732	0.6621179	-0.7088884	*	1	17	145	255	17	14
4	sp_6_sp_3.dat	3	0	50	-0.4330127	0.6246954	0.5106882	*	-0.3429557	1.1676334	0.5728124	*	1	5	145	119	5	14

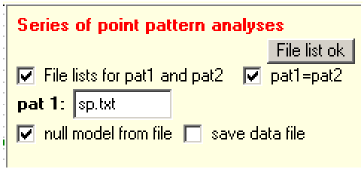
where T0(K12) 1 is the effect size of the $K_{12}(r)$ at distance $r = 1$ and T0(D12) 1 the effect size of $D_{12}(r)$ at distance $r = 1$ and so on.



21. Based on the values of the *M* and *P* axes that are given as effect sizes, you can determine the association type of a given species pair. Because we use two test statistics at the same time [$L_{ij}(r)$, $D_{ij}(r)$] we need to use a P-value of 0.025 for each summary functions to yield an overall error rate of 5%. That means that values of the effect size > 2.24 or < – 2.24 indicate a significant departure from the null model with the types shown on the left. Nicely, now the non-significant associations are all placed in the blue square in the middle, and the farther away from the square a species pair is located the stronger is the departure from the null model.

Instructions for the scheme based on univariate component patterns and null models from file

You can do the same procedure as above also for cases where you previously saved the null model files for the different patterns listed in the file lists, for example generated with **pattern reconstruction**.



Series of point pattern analyses

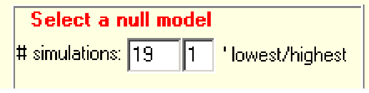
☑ File lists for pat1 and pat2 ☑ pat1=pat2

pat 1: sp.txt

☑ null model from file □ save data file

In this case you need to click "null model from file". The null model patterns corresponding to your data files must follow the name conventions:

data file: `name.dat`
null model file: `rec_name_n.dat`



Select a null model

simulations: 19 1 ' lowest/highest

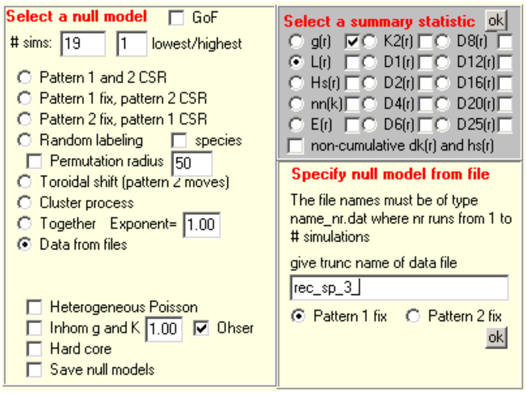
where `name` is the data file (e.g., sp_3 in the example from the file list above) and `n` the number that should run from 1 to the number of # simulations of the null model specified in the window "**Select a null model**".

Derive scheme with pattern reconstruction null model (Example scheme_rec.res)

For each univariate data file you need to generate as much reconstructions as you select for the number of simulations of the null model.

In the first step you have to prepare one example analysis with the correct settings and then save the corresponding *.res file.

1. First, execute *Programita*.
2. Highlight data file sp_1_3.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
5. Press button "**Calculate Index**"
6. Select L(r) in window "Select a summary function"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface. Select "**Data from file**" in the window "Select a null model".
8. Specify the **number of simulations of the null model** (19 in the example) and the rule for the estimation of **simulation envelopes** (here the 1th lowest and highest values of the summary function of the 19 null model data sets).
9. Provide trunk name of null model files in window "Specify null model from file". It is "rec_sp_3_" because the data file used sp_1.dat as focal pattern and sp_3.dat as second pattern. Select also "**Pattern 1 fix**" because you randomize pattern 2 and click small "**ok**" button.



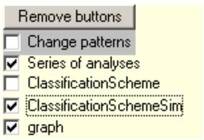
Select a null model □ GoF
# sims: 19 1 lowest/highest
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
● Data from files
□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

Select a summary statistic ok
○ g(r) ☑ ○ K2(r) □ ○ D8(r) □
● L(r) □ ○ D1(r) □ ○ D12(r) □
○ Hs(r) □ ○ D2(r) □ ○ D16(r) □
○ nn(k) □ ○ D4(r) □ ○ D20(r) □
○ E(r) □ ○ D6(r) □ ○ D25(r) □
□ non-cumulative dk(r) and hs(r)

Specify null model from file
The file names must be of type name_nr.dat where nr runs from 1 to # simulations
give trunc name of data file
rec_sp_3_
● Pattern 1 fix ○ Pattern 2 fix
ok

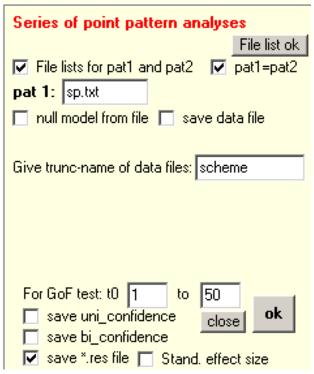
If all settings are specified, press the button "**Calculate Index**" and *Programita* conducts the simulations of the null model.

10. Save results file as "scheme_rec.res"
11. Enable check box "**ClassificationSchemeSim**" on the right hand side of the "Select a null model" window. A window opens with settings of the Series of analyses.



Remove buttons
□ Change patterns
☑ Series of analyses
□ ClassificationScheme
☑ ClassificationSchemeSim
☑ graph

12. To select files for pairwise analyses based on the univariate data files select "**File list for pat1 and pat2**".
13. If pattern 1 and 2 should be selected from the same list select "**Pat1=pat2**" (click the check box "**pat1=pat2**" two times). This is necessary to omit that the same file is selected as pattern 1 and pattern 2.
14. To save the bivariate data files enable "**save data file**"
15. Insert the name of the file list for **pat 1** (and if appropriate, for pat 2). In our case it is "sp.txt".
16. The trunk-name is used to name the summary output file. We use here the name "scheme"
17. Provide the distance interval for the GoF test (1-50m in our case)
18. Once all settings are specified, click the fat **ok** bottom ok , and then "**Calculate Index**" to start the series of analyses.
19. Disable the checkbox "graph" to simulate quicker.



Series of point pattern analyses
File list ok
☑ File lists for pat1 and pat2 ☑ pat1=pat2
pat 1: sp.txt
□ null model from file □ save data file
Give trunc-name of data files: scheme
For GoF test: t0 1 to 50
□ save uni_confidence close ok
□ save bi_confidence
☑ save *.res file □ Stand. effect size

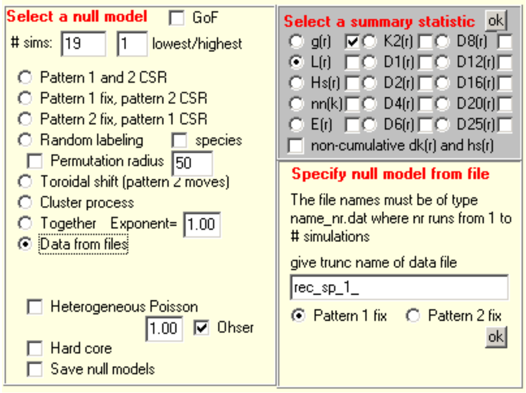
The scheme for numbered files and pattern reconstruction null model (Example scheme_nr_rec.res)

You can also run the classifications scheme analysis with numbered data files and the corresponding null model files from pattern reconstruction for the second pattern (the first pattern is unchanged).

In the example the data files are named “sp_6_sp_1.dat”, “sp_6_sp_2.dat”, and “sp_6_sp_3.dat”, or “sp_6_sp_nr.dat” where nr runs from 1 to 3.

The null model files which are only for the second species are named “rec_sp_1_n.dat”, “rec_sp_2_n.dat”, and “rec_sp_3_n.dat” where n runs from 1 to 19 (the number of simulations of the null model) and the blue marked number must corresponds to the nr of the data files.

1. First, execute *Programita*.
2. Highlight the first data file sp_6_sp_1.dat you want to analyze in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
5. Press button “**Calculate Index**”
6. Select L(r) in window “Select a summary function”
7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface. Select “**Data from file**” in the window “Select a null model”.
8. Specify the **number of simulations of the null model** (19 in the example) and the rule for the estimation of **simulation envelopes** (here the 1th lowest and highest values of the summary function of the 19 null model data sets).
   Provide trunk name of null model files in window “Specify null model from file”. It is “rec_sp_1_” because the data file used sp_1.dat as second pattern. Select also “**Pattern 1 fix**” because you randomize pattern 2 and click small “**ok**” button.

   Select a null model ☐ GoF
   # sims: 19 1 lowest/highest
   ○ Pattern 1 and 2 CSR
   ○ Pattern 1 fix, pattern 2 CSR
   ○ Pattern 2 fix, pattern 1 CSR
   ○ Random labeling ☐ species
   ☐ Permutation radius 50
   ○ Toroidal shift (pattern 2 moves)
   ○ Cluster process
   ○ Together Exponent= 1.00
   ● Data from files
   ☐ Heterogeneous Poisson
   1.00 ☑ Ohser
   ☐ Hard core
   ☐ Save null models

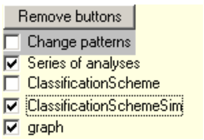
   Select a summary statistic ok
   ○ g(r) ☑ ○ K2(r) ☐ ○ D8(r) ☐
   ● L(r) ☐ ○ D1(r) ☐ ○ D12(r) ☐
   ○ Hs(r) ☐ ○ D2(r) ☐ ○ D16(r) ☐
   ○ nn(k) ☐ ○ D4(r) ☐ ○ D20(r) ☐
   ○ E(r) ☐ ○ D6(r) ☐ ○ D25(r) ☐
   ☐ non-cumulative dk(r) and hs(r)

   Specify null model from file
   The file names must be of type name_nr.dat where nr runs from 1 to # simulations
   give trunc name of data file
   rec_sp_1_
   ● Pattern 1 fix ○ Pattern 2 fix
   ok

   If all settings are specified, press the button “**Calculate Index**” and *Programita* conducts the simulations of the null model.
9. Save results file as “scheme_nr_rec.res”

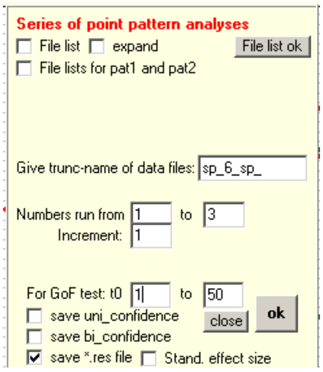
The classification scheme. Example *scheme_nr_rec.res*

10. Enable check box “**ClassificationSchemeSim**” on the right hand side of the “Select a null model” window. A window opens with settings of the Series of analyses.



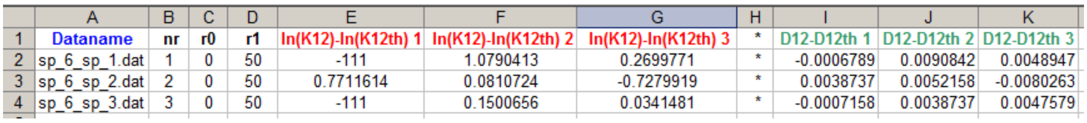
Remove buttons
☐ Change patterns
☑ Series of analyses
☐ ClassificationScheme
☑ ClassificationSchemeSim
☑ graph

11. Provide the trunk name for the data files (sp_6_sp_), and select the appropriate numbers because the number nr runs from 1 to 3 (i.e., you have 3 data files).
12. Select “save bi_confidence” and “D1(r)” to save the *.env files with names D12_1(r)_nr.env, Bi_confidencenr.env.
13. Provide the distance interval for the GoF test (1-50m in our case)
14. Once all settings are specified, click the fat **ok** bottom ok, and then “**Calculate Index**” to start the series of analyses.
15. Disable the checkbox “graph” to simulate quicker.

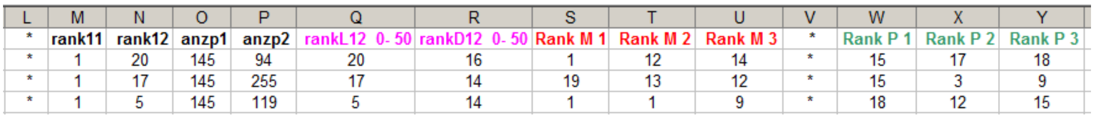


Series of point pattern analyses
☐ File list ☐ expand   File list ok
☐ File lists for pat1 and pat2
Give trunc-name of data files: sp_6_sp_
Numbers run from 1 to 3
Increment: 1
For GoF test: t0 1 to 50
☐ save uni_confidence
☐ save bi_confidence
☑ save *.res file ☐ Stand. effect size
close   ok

16. The output file “Summary_SchemeSim_scheme.txt” (“scheme” is the name you selected) is:



	A	B	C	D	E	F	G	H	I	J	K
1	Dataname	nr	r0	r1	ln(K12)-ln(K12th) 1	ln(K12)-ln(K12th) 2	ln(K12)-ln(K12th) 3	*	D12-D12th 1	D12-D12th 2	D12-D12th 3
2	sp_6_sp_1.dat	1	0	50	-111	1.0790413	0.2699771	*	-0.0006789	0.0090842	0.0048947
3	sp_6_sp_2.dat	2	0	50	0.7711614	0.0810724	-0.7279919	*	0.0038737	0.0052158	-0.0080263
4	sp_6_sp_3.dat	3	0	50	-111	0.1500656	0.0341481	*	-0.0007158	0.0038737	0.0047579



L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
*	rank11	rank12	anzp1	anzp2	rankL12 0- 50	rankD12 0- 50	Rank M 1	Rank M 2	Rank M 3	*	Rank P 1	Rank P 2	Rank P 3
*	1	20	145	94	20	16	1	12	14	*	15	17	18
*	1	17	145	255	17	14	19	13	12	*	15	3	9
*	1	5	145	119	5	14	1	1	9	*	18	12	15

4.2.9 Bivariate Thomas process with shared parents

The simple univariate Thomas process can be generalized to a bivariate point process that includes an explicit mechanism of attraction between the two component patterns.

While in a simple Thomas process the points of one pattern are distributed around the cluster centers, the bivariate Thomas process distributes the points of two patterns around the cluster centers. Each component pattern has an own parameter ρ determining the number of cluster centers for this pattern and a parameter σ determining the approximate size of the clusters. See section 4.2.3.1 “Bivariate Thomas Process with Shared Parents” in Wiegand and Moloney (2014). This cluster process was first presented in Jacquemyn et al. (2007).

If $\rho_1 = \rho_2$ all cluster centers are shared and the attraction between pattern 1 and 2 will be maximal. However, if $\rho_1 < \rho_2$ or $\rho_1 > \rho_2$ not all clusters host points of both types. In this case some parents are not shared. The fewer parents are shared, the less attraction exists between pattern 1 and pattern 2. Note that the degree of attraction is also determined by the size of the clusters.

The pair correlation function of the simple Thomas process yields:

$$g(r,\rho_1,\sigma_1) = 1 + \frac{1}{\rho_1} h_2(r,\sigma_1) = + \frac{1}{\rho_1} \frac{\exp(-r^2/4\sigma_1^2)}{4\pi\sigma_1^2}$$

and the pair correlation function of the simple bivariate Thomas process with shared parents (i.e., $\rho_1 = \rho_2 = \rho$; all parents are shared) yields

$$g_{12}(r,\rho_1,\sigma_1,\sigma_2) = 1 + \frac{1}{\rho} \frac{\exp(-r^2/[2(\sigma_1^2+\sigma_2^2)]}{2\pi(\sigma_1^2+\sigma_2^2)} = 1 + \frac{1}{\rho} \frac{\exp(-r^2/[4(\sigma_1^2+\sigma_2^2)/2]}{4\pi(\sigma_1^2+\sigma_2^2)/2}$$

Thus, *Programita* can fit basically the pair correlation function of a simple Thomas process to the observed bivariate pair correlation function, but with parameters $\sigma_t^2 = (\sigma_1^2 + \sigma_2^2)/2$ and ρ:

$$g_{12}(r,\rho,\sigma_t) = 1 + \frac{1}{\rho} \frac{\exp(-r^2/4\sigma_t^2)}{4\pi\sigma_t^2}$$

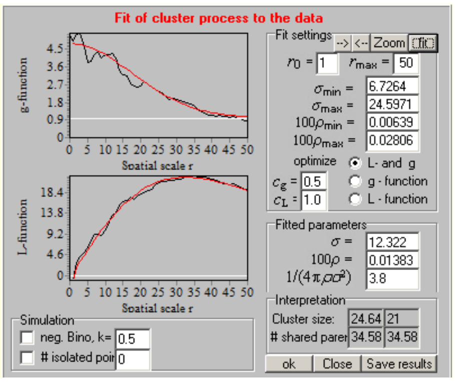
The fitted point process is therefore only appropriate if the fitted value of σ_t yields in good approximation $\sigma_t^2 = (\sigma_1^2 + \sigma_2^2)/2$ and if $\rho_1 \approx \rho_2 \approx \rho$. To test this, you need first to determine the parameters ρ_1, ρ_2, σ_1 and σ_2 of the univariate patterns using univariate analysis.

This example is based on the example Book_Fig4_13_bi.res and the two univariate patterns are taken from two different realizations of the univariate parent-offspring Thomas process. That means that the cluster centers are the same (i.e., shared parents) and the cluster sizes are also the same. The parameters of the original point process were σ = 12.54 and $A\rho$ = 34.3. Use a ring width of dr = 7 for the analyses.

Example BiThomasShared.res

1. In the first step fit a simple Thomas process to the two univariate component patterns to verify that the assumption of this point process (i.e., both patterns follow a simple Thomas process) holds.
2. The fitted parameters of pattern 1 (file BiThomasShared_p1.dat) yield σ_1 = 12.3 and $A\rho_1$ = 34.6
3. The fitted parameters of pattern 2 (file BiThomasShared_p2.dat) yield σ_2 = 13.6 and $A\rho_2$ = 28.4

pattern 1:



Fit of cluster process to the data

Fit settings

r_0 = 1 r_{max} = 50

σ_{min} = 6.7264

σ_{max} = 24.5971

$100\rho_{min}$ = 0.00639

$100\rho_{max}$ = 0.02806

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 12.322

100ρ = 0.01383

$1/(4\pi\rho\sigma^2)$ 3.8

Interpretation

Cluster size: 24.64 21

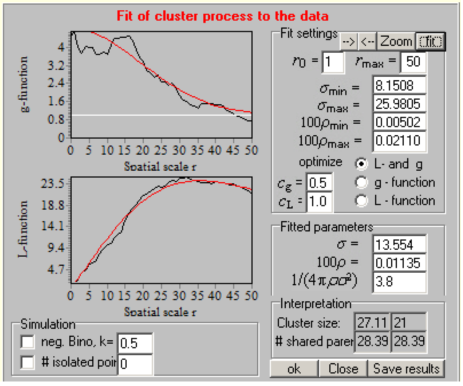
shared parer 34.58 34.58

Simulation

□ neg. Bino, k= 0.5

□ # isolated poir 0

pattern 2:



Fit of cluster process to the data

Fit settings

r_0 = 1 r_{max} = 50

σ_{min} = 8.1508

σ_{max} = 25.9805

$100\rho_{min}$ = 0.00502

$100\rho_{max}$ = 0.02110

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 13.554

100ρ = 0.01135

$1/(4\pi\rho\sigma^2)$ 3.8

Interpretation

Cluster size: 27.11 21

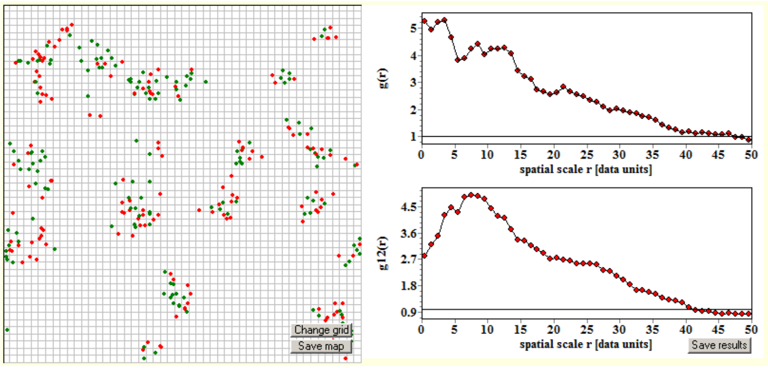
shared parer 28.39 28.39

Simulation

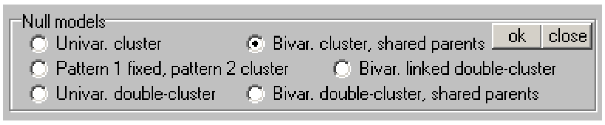
□ neg. Bino, k= 0.5

□ # isolated poir 0

11. Using the information from the univariate analysis, now fit in a second step the bivariate Thomas process with shared parents to the data
12. Execute *Programita*.
13. Highlight data file BiThomasShared.dat you want to analyze in Input data and click the small “ok” button.
14. Select bin of 1m window Select a new cell size
15. Select a ring width of 7 in the menu “Which method will you use”
16. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
17. Press button “**Calculate Index**”
18. *Programita* then shows the bivariate pattern. It is clear that the points of the two patterns are merged within the same clusters:



19. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
20. Select “**Cluster process**” in the window “Select a null model”.
21. A window “Fit of cluster process to data” opens. Select in the section “Null models” at the bottom the button “**Bivar. cluster, shared parents**”.



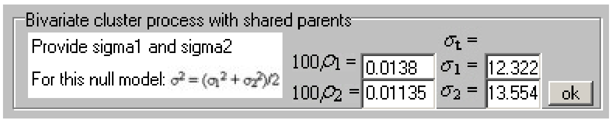
Null models

○ Univar. cluster ● Bivar. cluster, shared parents

○ Pattern 1 fixed, pattern 2 cluster ○ Bivar. linked double-cluster

○ Univar. double-cluster ○ Bivar. double-cluster, shared parents

Now this window appears that asks you to provide the parameters from the univariate analysis of patterns 1 and 2. Copy-paste the results from the univariate analyses and press the small “**ok**” buttons:



Bivariate cluster process with shared parents

Provide sigma1 and sigma2

For this null model: $\sigma^2 = (\sigma_1^2 + \sigma_2^2)/2$

$100\rho_1$ = 0.0138

$100\rho_2$ = 0.01135

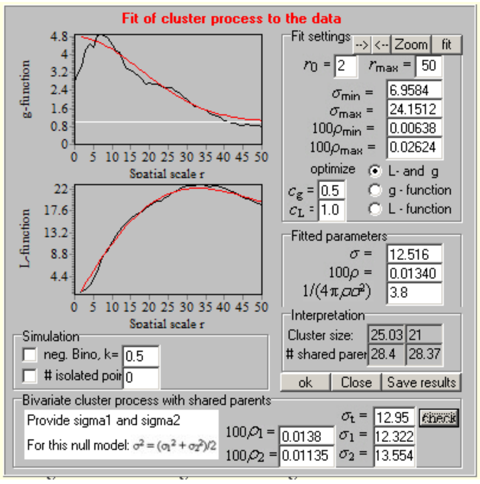
σ_t =

σ_1 = 12.322

σ_2 = 13.554

22. Now you are at the fitting window. Use the “**fit**” and “**zoom**” buttons to find a good fit. *Programita* fits the bivariate pair correlation function to the data, estimating

$$g_{12}(r,\rho,\sigma_t) = 1 + \frac{1}{\rho}\frac{\exp(-r^2/4\sigma_t^2)}{4\pi\sigma_t^2} \text{ where } \sigma_t^2 = (\sigma_1^2 + \sigma_2^2)/2\,.$$



Fit of cluster process to the data

Fit settings

r_0 = 2 r_{max} = 50

σ_{min} = 6.9584

σ_{max} = 24.1512

$100\rho_{min}$ = 0.00638

$100\rho_{max}$ = 0.02624

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 12.516

100ρ = 0.01340

$1/(4\pi\rho\sigma^2)$ 3.8

Interpretation

Cluster size: 25.03 21

shared parer 28.4 28.37

Simulation

□ neg. Bino, k= 0.5

□ # isolated poir 0

Bivariate cluster process with shared parents

Provide sigma1 and sigma2

For this null model: $\sigma^2 = (\sigma_1^2 + \sigma_2^2)/2$

$100\rho_1$ = 0.0138

$100\rho_2$ = 0.01135

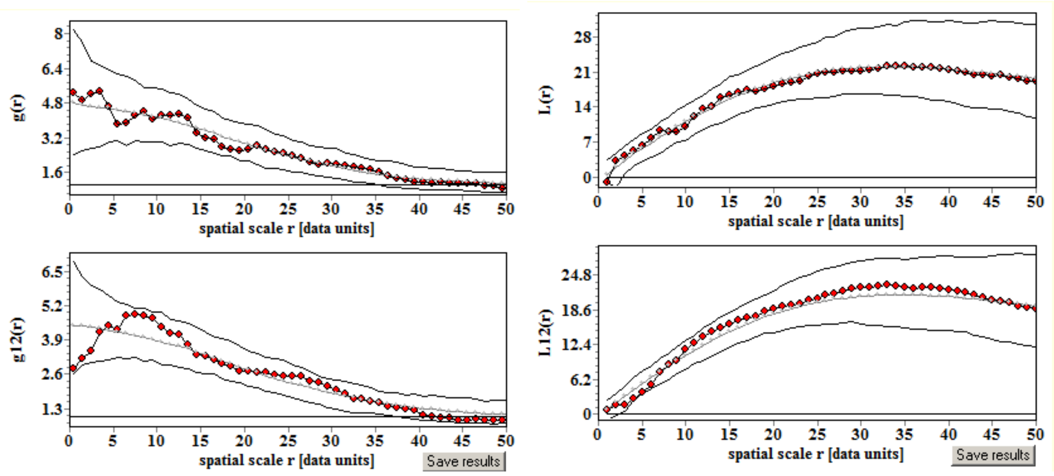
σ_t = 12.95

σ_1 = 12.322

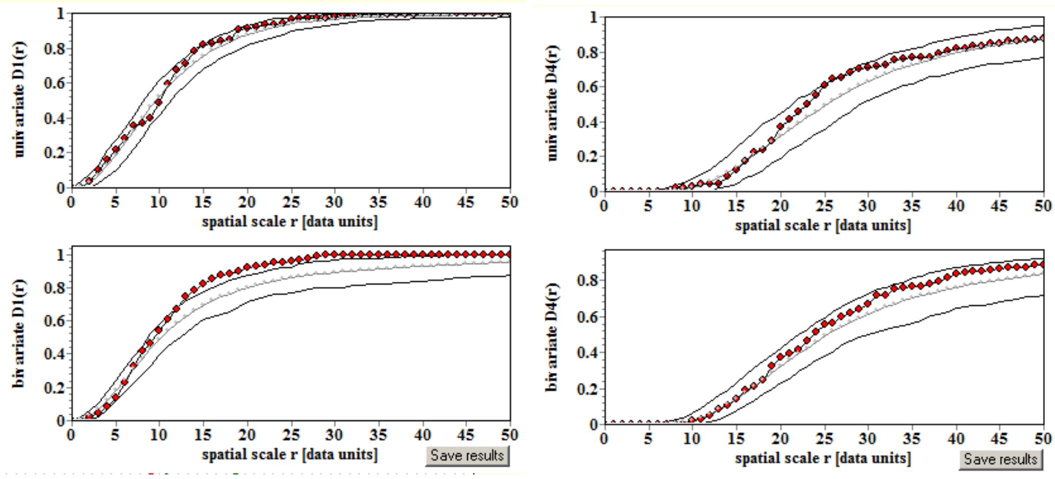
σ_2 = 13.554

The best fit yields a parameter *Aρ* = 28.4 shared parents and the fitted value of σ_t = 12.51 is in good agreement with $\sigma_t = ((\sigma_1^2 + \sigma_2^2)/2)^{0.5}$ = 12.95 (click “check” for estimating σ_t. Then click the “ok” button and “Calculate Index”.

23. As expected, the different summary functions are in good agreement with the fitted point process for both, the univariate analysis of pattern 1 and the bivariate analysis:



The weak departure in the bivariate nearest neighbor distribution function is caused by the underestimation of the number of clusters in pattern 2. A few cluster centers were not shared and therefore for the non-shared clusters of pattern 1 the nearest type 2 neighbors were farther away than expected from the data:



4.2.10 Bivariate Thomas process with partly shared parents

Not all clusters of the bivariate Thomas process with shared clusters need to be shared. If fewer parents are shared, the attraction of the two patterns will decline and in the extreme case where no cluster is shared they will be independent. Section 4.2.3.2 "Bivariate Thomas Process with Partly Shared Parents" in Wiegand and Moloney (2014) estimate the pair correlation function for the more general cluster process where not all clusters are shared. This cluster process was first presented in Jacquemyn et al. (2007). The pair correlation function of the simple bivariate Thomas process with partly shared parents yields

$$g_{12}(r, \rho^*, \sigma_1, \sigma_2) = 1 + \frac{1}{\rho^*} h_2(r, \sigma^*) \text{ where } \sigma^* = \sqrt{(\sigma_1^2 + \sigma_2^2)/2} \text{ and } \rho^* = \frac{\rho_1 \rho_2}{\rho_s}$$

The ρ_1 and ρ_2 are the fitted parameter of the univariate analyses of pattern 1 and 2, respectively, and the number of shared clusters yields $A\rho_s$. Thus, this point process has an "effective" number of clusters ρ^* which agrees with the number of clusters if all clusters are shared (i.e., $\rho_1 = \rho_2 = \rho_s$), and which becomes very large if no clusters are shared (i.e., ρ_s is small).

Thus, as before *Programita* fits basically the pair correlation function of a simple Thomas process to the observed bivariate pair correlation function and estimates ρ^* and σ^*.

Comparing the values of ρ^* and σ^* with the values of the parameters from the univariate analysis σ_1, $A\rho_1$, σ_2, and $A\rho_2$ allows to find out if the point process yields consistent parameters. First, the fitted value for σ^* should yield in approximation $\sigma^* = ((\sigma_1^2 + \sigma_2^2)/2)^{0.5}$. Second, we also expect $\rho_1 > \rho_s$ and $\rho_2 > \rho_s$, because the number of shared parents cannot be greater than the number of parents of pattern 1 or 2.

Bivariate Thomas process with shared parents. Example *Book_Fig4_29.res*

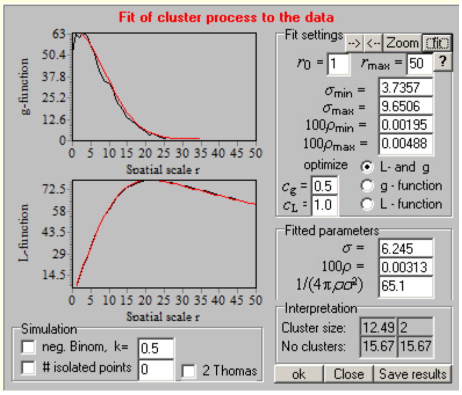
This example is based on a realization of the bivariate Thomas process with partly shared parents fitted in example of Figure 4.29 in Wiegand and Moloney (2014).

The parameters of the original point process were σ_1 = 6.65, and $A\rho_1$ = 14.85, σ_2 = 4.42, $A\rho_2$ = 64.75, and the fitted parameter of the shared parents was $A\rho$ = 10.3.

Example Book_Fig4_29.res

1. In the first step fit a simple Thomas process to the two univariate component patterns to verify that the assumption of this point process (i.e., both patterns follow a simple Thomas process) holds.
2. The fitted parameters of pattern 1 (file Book_Fig4_29_p1.dat) yield σ_1 = 6.25 and $A\rho_1$ = 15.67
3. The fitted parameters of pattern 2 (file Book_Fig4_29_p2.dat) yield σ_2 = 4.26 and $A\rho_2$ = 65.0

pattern 1:



Fit of cluster process to the data

Fit settings

r_0 = 1 r_{max} = 50

σ_{min} = 3.7357

σ_{max} = 9.6506

$100\rho_{min}$ = 0.00195

$100\rho_{max}$ = 0.00488

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 6.245

100ρ = 0.00313

$1/(4\pi\rho\sigma^2)$ 65.1

Interpretation

Cluster size: 12.49 2

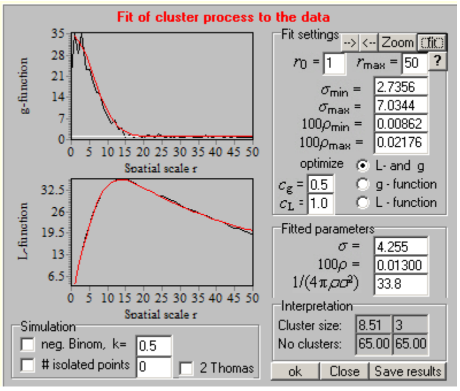
No clusters: 15.67 15.67

Simulation

□ neg. Binom, k= 0.5

□ # isolated points 0 □ 2 Thomas

pattern 2:



Fit of cluster process to the data

Fit settings

r_0 = 1 r_{max} = 50

σ_{min} = 2.7356

σ_{max} = 7.0344

$100\rho_{min}$ = 0.00862

$100\rho_{max}$ = 0.02176

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 4.255

100ρ = 0.01300

$1/(4\pi\rho\sigma^2)$ 33.8

Interpretation

Cluster size: 8.51 3

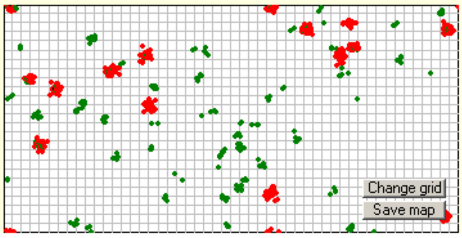
No clusters: 65.00 65.00

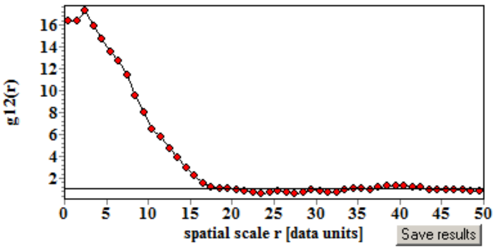
Simulation

□ neg. Binom, k= 0.5

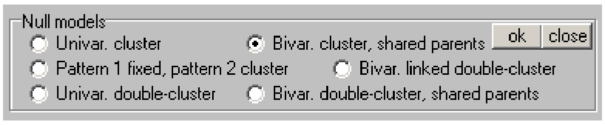
□ # isolated points 0 □ 2 Thomas

4. Using the information from the univariate analysis, now fit in a second step the bivariate Thomas process with shared parents to the data
5. Execute *Programita*.
6. Highlight data file Book_Fig4_29.dat you want to analyze in Input data and click the small “ok” button.
7. Select bin of 1m window Select a new cell size
8. Select a ring width of 3 in the menu “Which method will you use”
9. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
10. Press button “**Calculate Index**”
11. *Programita* then shows the bivariate pattern. It is clear that not all clusters are shared but that there is a strong attraction between the two patterns:

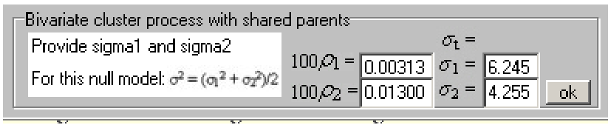




12. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
13. Select “**Cluster process**” in the window “Select a null model”.
14. A window “Fit of cluster process to data” opens. Select in the section “Null models” at the bottom the button “**Bivar. cluster, shared parents**”.

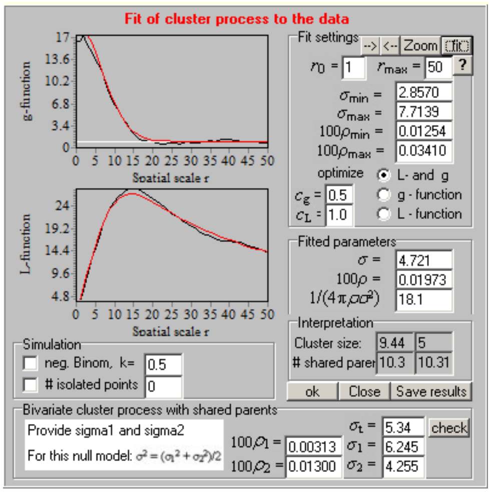

Null models
○ Univar. cluster ● Bivar. cluster, shared parents
○ Pattern 1 fixed, pattern 2 cluster ○ Bivar. linked double-cluster
○ Univar. double-cluster ○ Bivar. double-cluster, shared parents

Now this window appears that asks you to provide the parameters from the univariate analyses of patterns 1 and 2. Copy-paste the results from the univariate analyses and press the small “**ok**” buttons:


Bivariate cluster process with shared parents
Provide sigma1 and sigma2
For this null model: $\sigma^2 = (\sigma_1^2 + \sigma_2^2)/2$
σ_t =
$100\rho_1$ = 0.00313 σ_1 = 6.245
$100\rho_2$ = 0.01300 σ_2 = 4.255

15. Now you are at the fitting window. Use the “fit” and “zoom” buttons to find a good fit. *Programita* fits the bivariate pair correlation function to the data, estimating

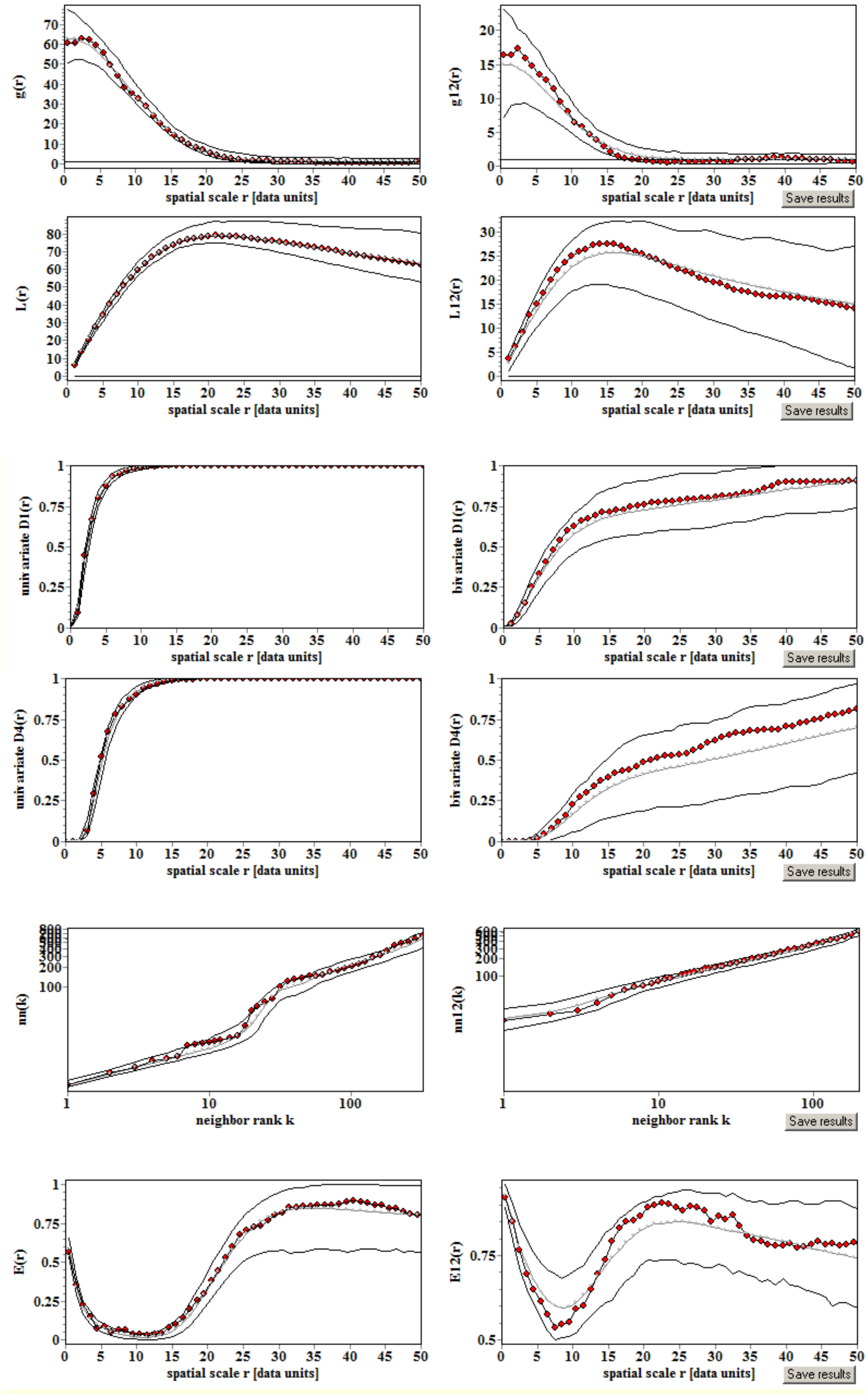
$$g_{12}(r,\rho,\sigma_t) = 1 + \frac{1}{\rho}\frac{\exp(-r^2/4\sigma_t^2)}{4\pi\sigma_t^2} \quad \text{where } \sigma_t^2 = (\sigma_1^2 + \sigma_2^2)/2\,.$$


Fit of cluster process to the data
Fit settings
r_0 = 1 r_{max} = 50
σ_{min} = 2.8570
σ_{max} = 7.7139
$100\rho_{min}$ = 0.01254
$100\rho_{max}$ = 0.03410
optimize ● L- and g ○ g - function ○ L - function
c_g = 0.5
c_L = 1.0
Fitted parameters
σ = 4.721
100ρ = 0.01973
$1/(4\pi\rho\sigma^2)$ 18.1
Interpretation
Cluster size: 9.44 5
# shared parer 10.3 10.31
Simulation
□ neg. Binom, k= 0.5
□ # isolated points 0
Bivariate cluster process with shared parents
Provide sigma1 and sigma2
For this null model: $\sigma^2 = (\sigma_1^2 + \sigma_2^2)/2$
σ_t = 5.34
$100\rho_1$ = 0.00313 σ_1 = 6.245
$100\rho_2$ = 0.01300 σ_2 = 4.255

The best fit yields a parameter *Aρ* = 10.3 shared parents and the fitted value of σ_t = 4.7 is in good agreement with $\sigma_t = ((\sigma_1^2 + \sigma_2^2)/2)^{0.5}$ = 5.34.

16. As expected, the different summary functions are in good agreement with the fitted point process for both, the univariate analysis of pattern 1 and the bivariate analysis. However, note the relatively wide pointwise simulation envelopes of the bivariate analysis which indicates a strong stochastic variability among the realizations of this point process.

Bivariate Thomas process with shared parents. Example *Book_Fig4_29.res*



4.2.11 Bivariate hard and soft core processes

Gibbs (or Markov) point processes can be used to consider interaction among points of one type and between points of different type (see section 4.2.3.4 “Gibbs Processes to Model Interactions between Species” in Wiegand and Moloney 2014). They are natural generalizations of the univariate Gibbs processes.

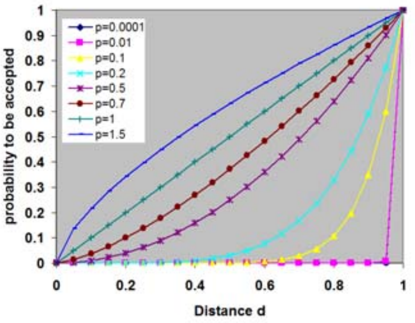
Simulation of bi (or multivariate) Gibbs processes requires optimization techniques where points of an initial pattern are deleted and replaced by randomly drawn points, which are accepted if the new point configuration becomes more likely, given the location density function. Such “birth and death” simulation algorithms closely resemble aspects of spatial population or community dynamics and are structurally very similar to spatially explicit, individual-based simulation models (Grimm and Railsback 2005) which are used by ecologists to study the spatiotemporal dynamics of plant populations and communities.

Programita has not implemented Gibbs processes, for ecologists it is recommend to use instead individual-based models that are based on direct biological mechanisms. However, to provide you the possibility to simulate simple point patterns with repulsion or segregation, *Programita* includes a simple algorithm based on “random sequential absorption” (RSA) processes to produce bivariate hard and soft core patterns.

The RSA algorithm implemented in *Programita* is simple. It is constructed by placing iteratively and randomly points within an observation window *W* which are thought to be the centers of disks with radius r_0. For bivariate patterns *Programita* first places the points of pattern 1 and then in a second step, the points of pattern 2. Thus, points of pattern 2 do not influence the placement of points of pattern 1.

Five parameters govern this point process, two radiuses r_1 and r_2 of pattern 1 and 2, respectively, that give the “zone of influence” of the two patterns, and three interaction coefficients p_1, p_2, and p_{21} that determine together with the distance d to the nearest already placed point the probability that the new point is accepted. As in the univariate case, this probability yields

$$p_{\mathrm{HC}}(d) \;=\; \begin{cases} d^{1/p} & \text{for } d \le r_{12} \\ 1 & \text{for } d > r_{12} \end{cases}$$



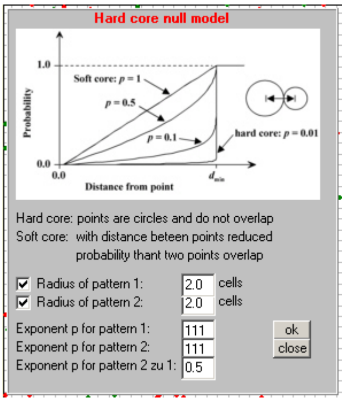
where p is p_1, p_2, and p_{12} depending on the types of the tentatively placed point and the nearest neighbor, $r_{12} = 2\ r_1$ if the tentatively placed point and the nearest neighbor are of type 1, $r_{12} = 2\ r_2$ if the tentatively placed point and the nearest neighbor are of type 2, and $r_{12} = r_1 + r_2$ if the tentatively placed point and the nearest neighbor are of different type.

This setting allows for different types of patterns with interactions only between points of type 1 and type 2, only between points of type 1, between points of type 2, and so on.

Example Book_Fig4_21e_bi.res (bivariate RSA inhibition process)

This pattern has been generated with a RSA algorithm to simulate non-overlapping disks with radius r_0 = 2m, but overlap was only restricted between disks of different type.

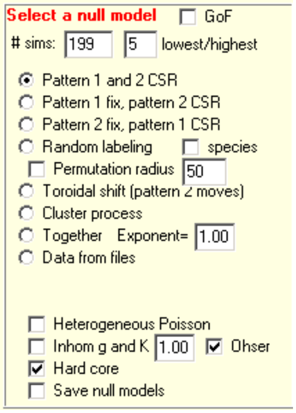
1. Execute *Programita*.
17. Highlight data file Book_Fig4_21e.dat you want to analyze in Input data and click the small “ok” button.
2. Select bin of 1m window Select a new cell size
3. Select a ring width of 3 in the menu “Which method will you use”
4. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
5. Press button “**Calculate Index**”
6. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
7. Select null model “Pattern 1 and 2 CSR” to start with the basic CSR algorithm.
8. Click checkbox “Hard core” and go to the window “Hard core null model” to define details of the RSA null model. Click “**Radius of pattern 1**” and “**Radius of pattern 2**”because you have a bivariate pattern and provide the radius (2.0) in our case.


Hard core null model

Hard core: points are circles and do not overlap
Soft core: with distance beteen points reduced
probability thant two points overlap

☑ Radius of pattern 1: 2.0 cells
☑ Radius of pattern 2: 2.0 cells

Exponent p for pattern 1: 111
Exponent p for pattern 2: 111
Exponent p for pattern 2 zu 1: 0.5

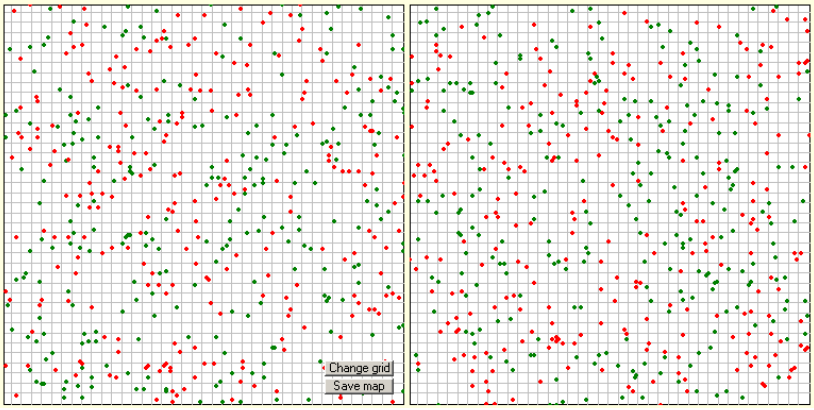

Select a null model □ GoF

sims: 199 5 lowest/highest

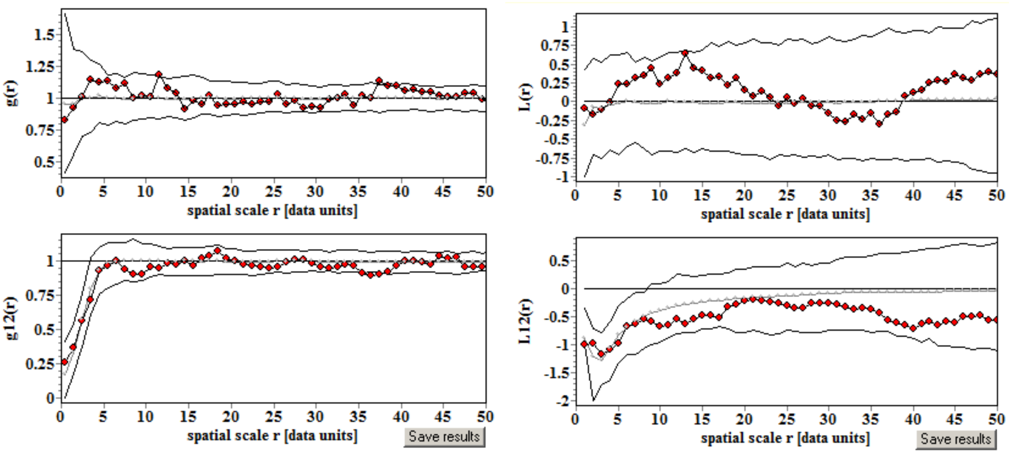
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

□ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
☑ Hard core
□ Save null models

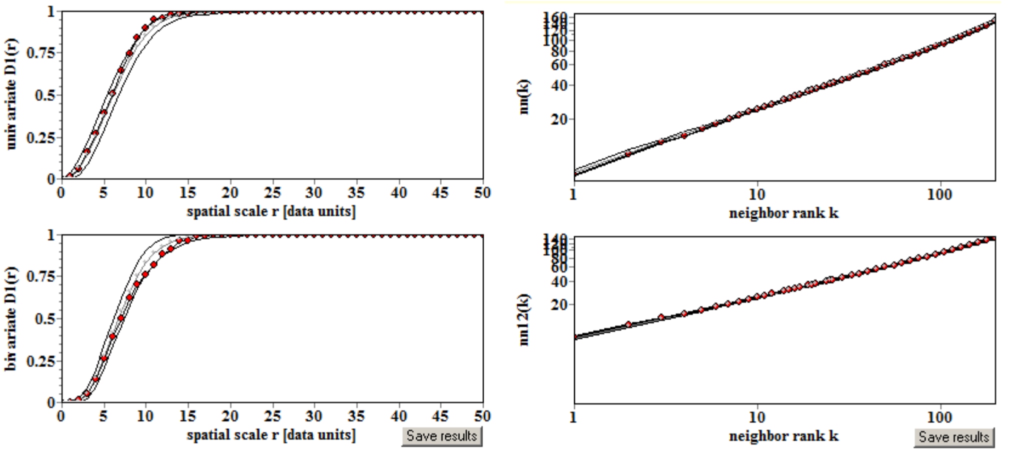
9. Because the two univariate component patterns should be CSR patterns, select large values of the exponents p_1 and p_2 e.g., 111. However, because of a negative interaction between type 1 and type 2 points select p_{21} = 0.5. Finally, click the small “**ok**” button.
10. To simulate the point process press “**Calculate Index**”. As expected, the simulated patterns look very similar to the observed pattern:



11. The uni- and bivariate pair correlation functions and the *L*-functions agree well with the simulated point process and the bivariate summary functions show the typical “soft-core” shape:



12. The same is true for the nearest neighbor distribution functions:

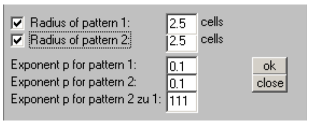


Example Book_Fig4_21a_bi.res (Bivariate inhibition process)

This pattern has been generated with a RSA algorithm to simulate non-overlapping disks with radius r_0 = 2m where “interactions” occurred only among points of the same type.

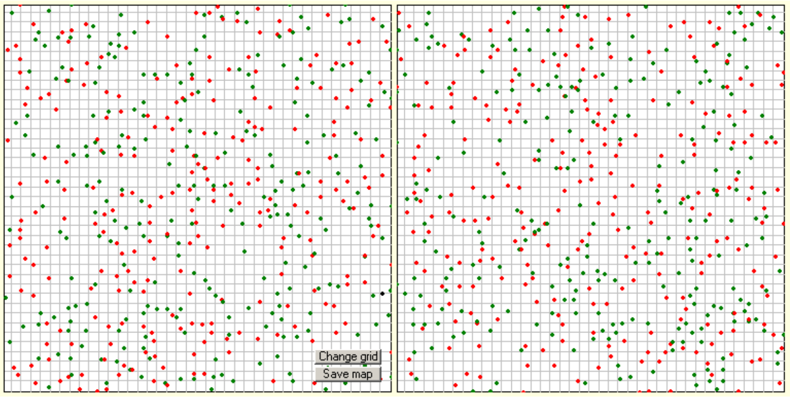
1. Execute *Programita*.
2. Highlight data file Book_Fig4_21a.dat you want to analyze in Input data and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button “**Calculate Index**”

7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
8. Select null model “Pattern 1 and 2 CSR” to start with the basic CSR algorithm.
9. Click checkbox “Hard core” and go to the window “Hard core null model” to define details of the RSA null model. Click “**Radius of pattern 1**” and “**Radius of pattern 2**”because you have a bivariate pattern and provide the radius (2.5) in our case.
10. Because the two univariate component patterns should be hard core patterns, select small values of the exponents p_1 and p_2 e.g., 0.1. However, because there should be no interaction between type 1 and type 2 points select p_{21} = 111. Finally, click the small “**ok**” button.

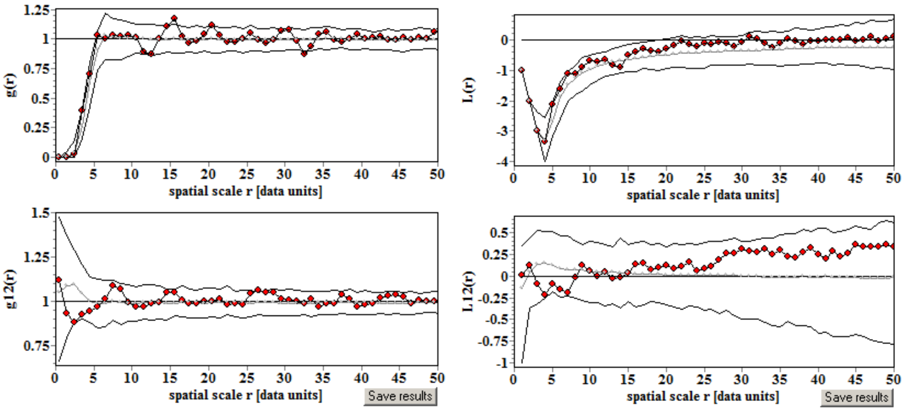


☑ Radius of pattern 1: 2.5 cells
☑ Radius of pattern 2: 2.5 cells
Exponent p for pattern 1: 0.1
Exponent p for pattern 2: 0.1
Exponent p for pattern 2 zu 1: 111

11. To simulate the point process press “**Calculate Index**”. As expected, the simulated patterns look very similar to the observed pattern:



12. The uni- and bivariate pair correlation functions and the *L*-functions agree well with the simulated point process, and the univariate summary functions show the typical “hard-core” shape:



The distribution functions of the distances to the nearest neighbor are also very well matched.

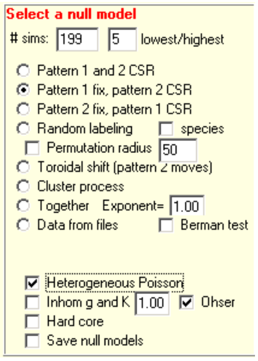
4.2.12 Heterogeneous Poisson processes for bivariate patterns

The heterogeneous Poisson process can be used to account for first-order heterogeneity [i.e., the pattern has a non-constant intensity function $\lambda(\mathbf{x})$] in one of the two component patterns of a bivariate pattern. The summary functions are still impacted by the heterogeneity, but the null model shows the same heterogeneity than the original pattern. The shape of the observed summary functions, relative to that of the simulated null model patterns, allows revealing potential effects of species interactions.

The following examples show simple cases of heterogeneity which are nevertheless often close to real-world patterns of tropical forests. In the examples the two component patterns are CSR and independent of each other, but only distributed within subareas A_1 and A_2 of the observation window W. Depending on the size and the overlap of the two subareas, the different association types mixing, partial overlap and segregation already introduced above in the classification scheme can occur.

Example Book_Fig4_31a.res
The following example presents the analysis of Figure 4.31 using a heterogeneous Poisson process with non-parametric kernel estimate for pattern 2 whereas pattern 1 remains unchanged. You can repeat the analyses with the data files Book_Fig4_31b.dat and Book_Fig4_31c.dat

1. Execute *Programita*.
2. Highlight data file Book_Fig4_31a.dat you want to analyze in Input data and click the small "ok" button
3. Select "**no grid**" in What do you want to do?
4. Select a ring width of 5 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Pattern 1 fix, pattern 2 CSR**" in the window "Select a null model".
9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5^{th} lowest and highest values of the summary function of the 199 simulated null model data sets).
10. Click checkbox "**Heterogeneous Poisson**"
11. Go to window "Settings for hetero. Poisson" on the left and insert the bandwidth R (30m in the example), enable "**Kernel**" for the Epanechnikov kernel and select "**Intensity of pattern 2**" (because this null model randomizes pattern 2). Edge correction "**Edge**" is enabled by default.
12. Click "**Calculate Index**" and *Programita* estimates the intensity function and shows the pattern and the corresponding intensity function.

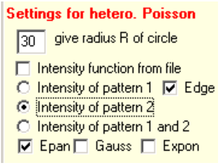


Select a null model

sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
● Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test

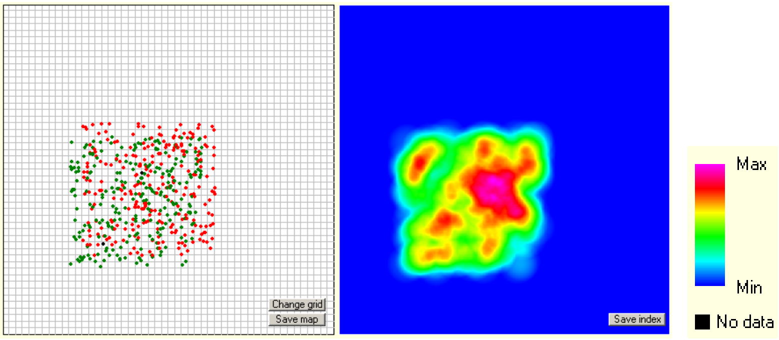
☑ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models



Settings for hetero. Poisson

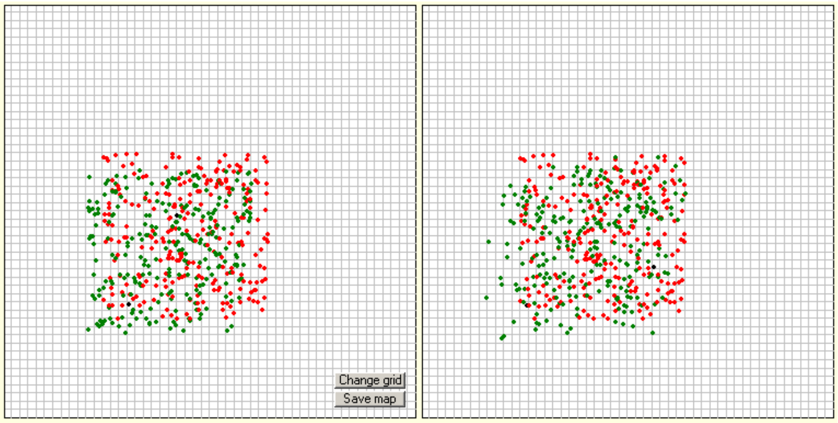
30 give radius R of circle

□ Intensity function from file
○ Intensity of pattern 1 ☑ Edge
● Intensity of pattern 2
○ Intensity of pattern 1 and 2
☑ Epan □ Gauss □ Expon



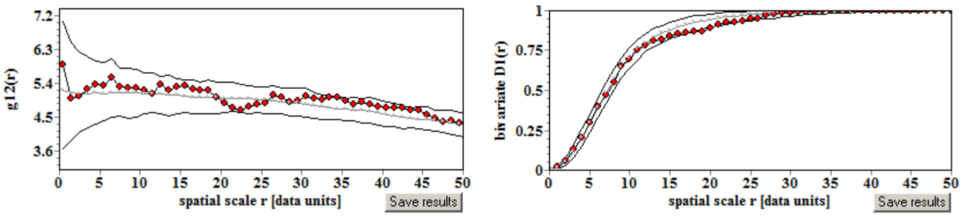
Click OK at the message box to save the intensity file. The file is saved with name int_E_Book_Fig4_31a_R2_30.int where the "int_E" indicates Epanechnikov kernel, Book_Fig4_31a.dat was the data file, "_R2_30" means that the intensity was estimated with pattern 2 and bandwidth 30.

13. Now *Programita* conducts the analysis. You can observe during the simulations that the null model distributes the points of pattern 2 (green points) with probability proportionally to the intensity. Here an example:



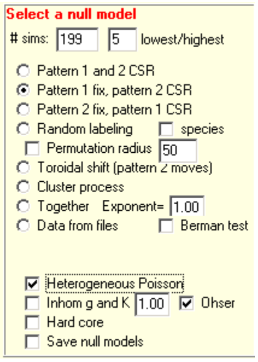
The strong heterogeneity of the second pattern (green points) is conserved, although pattern 2 is somewhat "smeared" because of the kernel function.

14. The result resembles that in Figure 4.31d, g well:

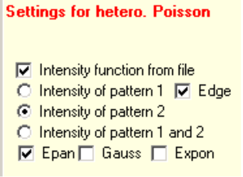


Example Book_Fig4_31a_int2.res
The following example presents the analysis above, but uses instead of the approximation of the intensity function based on a kernel estimate the "real" intensity function which has a value of λ inside the (100, 300) × (100, 300) subarea occupied by pattern 2. The null model again is a heterogeneous Poisson process for pattern 2 whereas pattern 1 remains unchanged.

1. Execute *Programita*.
2. Highlight data file Book_Fig4_31a.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 5 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Pattern 1 fix, pattern 2 CSR**" in the window "Select a null model".

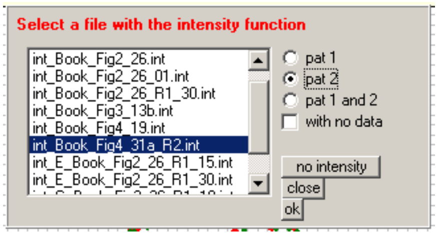


Select a null model
# sims: 199 5 lowest/highest
○ Pattern 1 and 2 CSR
● Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test
☑ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models



Settings for hetero. Poisson
☑ Intensity function from file
○ Intensity of pattern 1 ☑ Edge
● Intensity of pattern 2
○ Intensity of pattern 1 and 2
☑ Epan □ Gauss □ Expon

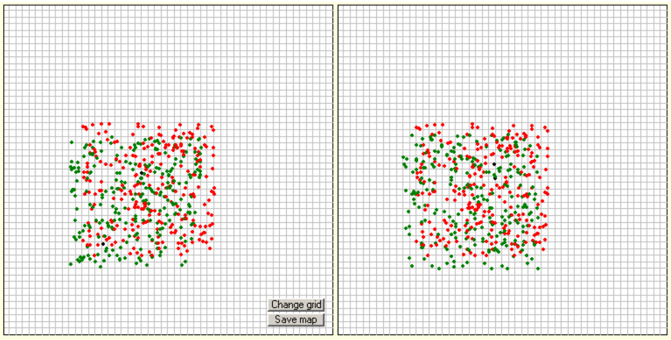
9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
10. Click checkbox "**Heterogeneous Poisson**"
11. Go to window "Settings for hetero. Poisson" on the left and select "**Intensity of pattern 2**" (because this null model randomizes pattern 2).
12. Click checkbox "**Intensity function from file**". The window "Select a file with the intensity function" opens.
13. Highlight in this window the file int_Book_Fig4_31a_R2.int that contains the intensity function,



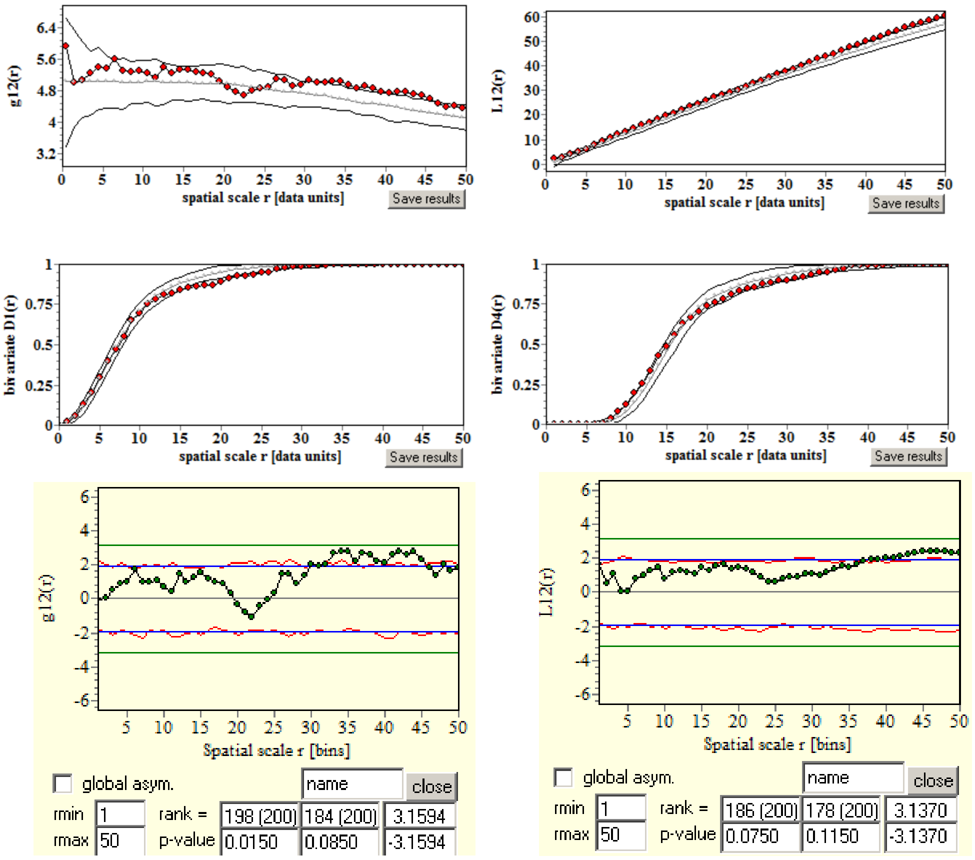
Select a file with the intensity function
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_Book_Fig2_26_R1_30.int
int_Book_Fig3_13b.int
int_Book_Fig4_19.int
int_Book_Fig4_31a_R2.int
int_E_Book_Fig2_26_R1_15.int
int_E_Book_Fig2_26_R1_30.int
○ pat 1
● pat 2
○ pat 1 and 2
□ with no data

select also "pat 2" because it is the intensity of pattern2 and then the small "**ok**" button. Be sure that "**Intensity of pattern 2**" is selected in the window "Settings for hetero. Poisson".

14. Click “**Calculate Index**” and *Programita* simulates the heterogeneous Poisson process. Because now the real intensity function was used, the null model patterns are not smeared as in the heterogeneous Poisson process with a kernel estimate of the intensity function:



15. The results are similar to the previous case, as expected, there is no significant departure from the null model although a slight (non-significant) departure is visible at larger scales in the *g*- and *L*-function:



□ global asym. name
rmin 1 rank = 198 (200) 184 (200) 3.1594
rmax 50 p-value 0.0150 0.0850 -3.1594

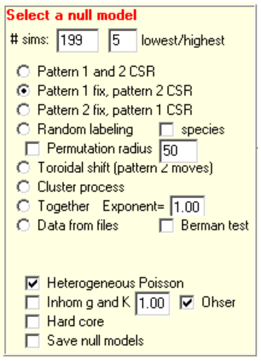
□ global asym. name
rmin 1 rank = 186 (200) 178 (200) 3.1370
rmax 50 p-value 0.0750 0.1150 -3.1370

4.2.13 Heterogeneous Poisson processes process with dispersal kernel

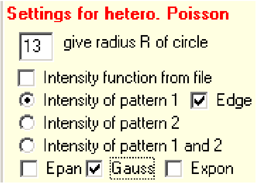
The example Book_Fig4_18.res was used in the univariate section ("Simple bivariate parent-offspring Thomas process") to illustrate the duality between a Cox process and a parent-offspring Thomas process where the cluster centers (i.e., parents) are known and are the same in each simulation of the point process. In this case we assume a "dispersal kernel" around the points of type 1 (which are the cluster centers) and the points of type 2 are distributed in accordance with this dispersal kernel. The advantage of the more flexible interpretation as Cox process (i.e., heterogeneous Poisson process) is that we can use additional kernel functions and not only the normal distribution as assumed in the Thomas process. Additionally, as shown in Rodríguez-Pérez et al. (2012), we can manipulate the intensity function to accommodate various hypotheses.

The procedure for this null model is the same as in the example above, but now you select in the window "Settings for hetero. Poisson "**Intensity of pattern 1**" (because this null model uses the intensity function of pattern 1 to randomize pattern 2).

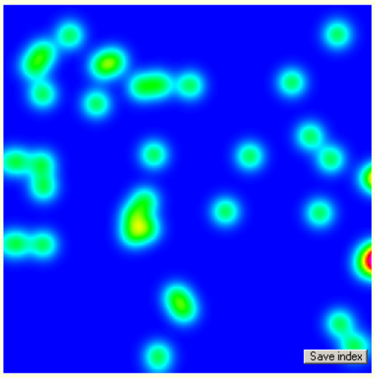
1. Execute *Programita*.
2. Highlight data file Book_Fig4_13.dat you want to analyze in Input data and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 7 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Select "**Pattern 1 fix, pattern 2 CSR**" in the window "Select a null model".
9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
10. Click checkbox "**Heterogeneous Poisson**"
11. Go to window "Settings for hetero. Poisson" on the left and insert the bandwidth R (13m in the example), enable "**Gauss**" for the kernel function being 2-dimensional normal distribution and select "**Intensity of pattern 1**" (because in this null model the intensity is based on pattern 1). Edge correction "**Edge**" is enabled by default.
12. You can also select alternative kernel functions such as the Epanechnikov kernel or the exponential kernel (see univariate section "Heterogeneous Poisson with kernel estimate".
13. Click "**Calculate Index**" and *Programita* estimates the intensity function and shows the pattern and the corresponding intensity function:



Select a null model
# sims: 199 5 lowest/highest
○ Pattern 1 and 2 CSR
● Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test
☑ Heterogeneous Poisson
□ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

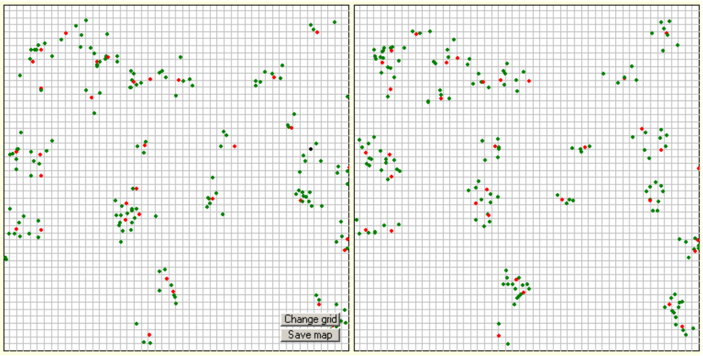


Settings for hetero. Poisson
13 give radius R of circle
□ Intensity function from file
● Intensity of pattern 1 ☑ Edge
○ Intensity of pattern 2
○ Intensity of pattern 1 and 2
□ Epan ☑ Gauss □ Expon

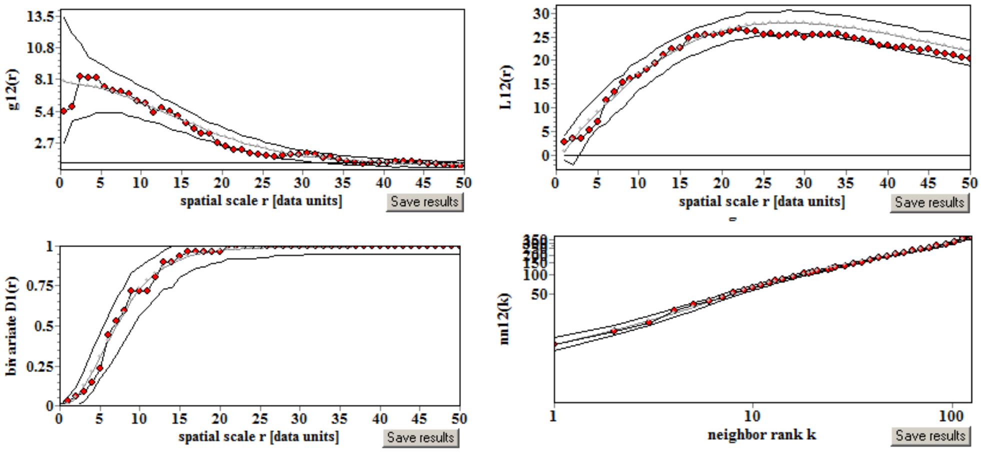


14. Click OK at the message box if you want to save the intensity file. The file is saved with int_G_Book_Fig4_18_R1_13.int where the “int_G” indicates Gaussian kernel, Book_Fig4_18.dat was the data file, “_R1_13” means that the intensity was estimated with pattern 1 and bandwidth 13. As shown in Rodríguez-Pérez et al. (2012), the intensity function can be further anipulated to accommodate additional hypotheses. For example, if can be multiplied (weighted) with a habitat suitability index for points of pattern 2.

   Now *Programita* conducts the analysis. You can observe during the simulations that the null model distributes the points with probability proportionally to the intensity function. Here an example:



15. As expected, the summary functions of the point process agree well with that of the observed data and resemble that of Figure 4.18:



4.2.14 Bivariate inhomogeneous *g*- and *K* functions

Inhomogeneous second-order statistics can be used in *Programita* for two different purposes. First, they offer a natural estimator of the second-order summary functions **for irregularly shaped observation windows**. In this case the *.int file that defines the intensity function $\lambda(\mathbf{x})$ contains only the values "1" (location **x** inside the observation window) and "0" (**x** outside the observation window). Second, the second-order statistics can be used to **remove the effect of environmental heterogeneity**, given by the intensity function $\lambda(\mathbf{x})$. The inhomogeneous *g*- and *K* functions then quantify the residual clustering or overdispersion of the data, conditionally on $\lambda(\mathbf{x})$.

Note that *Programita* can only handle one intensity function which is then applied to both, the uni- and the bivariate second-order summary functions. Thus, cases where the two component patterns show different response to environmental heterogeneity cannot be handled (although you can use a null model where the first pattern is unchanged and then select the adapted WM estimator to consider the intensity of the second pattern).

You can use in *Programita* two different estimators of the bivariate inhomogeneous *g*- and *K*-functions that are generalizations of the Ohser estimator (see Wiegand and Moloney 2014: sections 3.1.4.3 and 3.1.4.4. The bivariate Ohser estimator for two univariate patterns with n_1 and n_2 points $\mathbf{x}_{1,\mathrm{i}}$ and $\mathbf{x}_{2,\mathrm{i}}$ of pattern 1 and 2, respectively, works in the same way as the univariate estimator. It uses the points $\mathbf{y}_\mathrm{i}$ of an auxiliary heterogeneous Poisson process of intensity function $\lambda(\mathbf{x})$ with *m* points:

$$\hat{g}^{O}_{12,\text{inhom}}(r,\lambda(\mathbf{x})) \approx \frac{(m-1)m}{n_1 n_2} \frac{\sum_{i=1}^{n}(\sum_{j=1}^{n} k(\|\mathbf{x}_{1,i}-\mathbf{x}_{2,j}\|-r)}{\sum_{i=1}^{m}(\sum_{j=1}^{m} k(\|\mathbf{y}_{i}-\mathbf{y}_{j}\|-r)} \tag{1}$$

Note that the generalized isotropized set covariance $\bar{\gamma}_W(r,\lambda(\mathbf{x}))$ needs to be calculated only once and can be used for all simulations of the null model. *Programita* offers also a version of this estimator based on adapted intensity estimators presented in Illian et al. (2008):

$$\hat{g}^{WM}_{12,\text{inhom}}(r,\lambda_2(\mathbf{x})) \approx \frac{m}{n-1} \frac{\sum_{i=1}^{n}(\sum_{j=1}^{n} k(\|\mathbf{x}_{1,i}-\mathbf{x}_{2,j}\|-r)}{\sum_{i=1}^{m}(\sum_{j=1}^{m} k(\|\mathbf{x}_{1,i}-\mathbf{y}_{j}\|-r)} \tag{2}$$

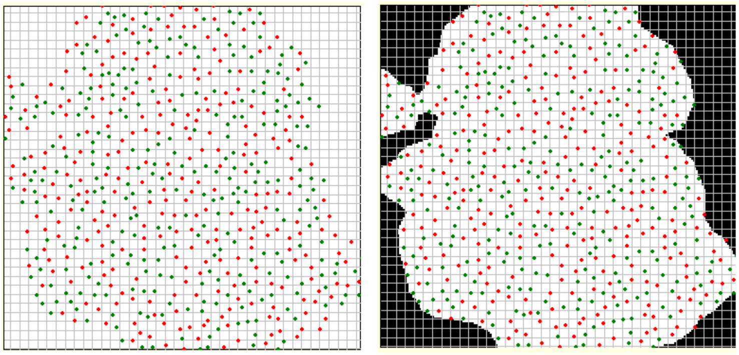
where only the second point in each point pair i–j is taken from the auxiliary pattern $\mathbf{y}_\mathrm{i}$. This estimator therefore compares the relative number of points $\mathbf{x}_{\mathrm{i},2}$ of pattern 2 in rings around the points $\mathbf{x}_{\mathrm{i},1}$ of pattern 1 with the relative number of points of the auxiliary pattern in rings around the points $\mathbf{x}_{\mathrm{i},1}$ of pattern 1. In contrast to the univariate case, the double-sum in the denominator must be evaluated only once if the null model keeps the points of pattern 1 fixed (i.e., an antecedent pattern).

Programita use a default value for the number *m* of points of the auxiliary heterogeneous Poisson process of

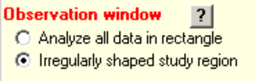
$m = 6000*6000/n_1$	(adapted WM estimator with pattern 1 not fixed)
$m = 20000*20000/n_1$	(adapted WM estimator with pattern 1 fixed)
$m = 30000$	(generalized Ohser estimator)

Example IrrInhomBi.res (bivariate pattern, irregular study area)

1. Execute *Programita*.
2. Highlight data file InhomBi.dat you want to analyze in Input data and click the small "ok" button. This pattern was created within an irregularly shaped observation window with a bivariate RSA inhibition process with "Radius of pattern 1" and "Radius of pattern 2" being 3 and exponents $p_1 = p_2 = p_{21} = 0.1$ that represent equal con- and heterospecific negative interactions within distance of 6m:



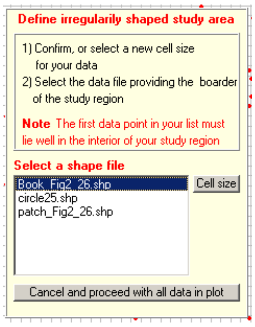
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the radio button "**Irregularly shaped study region**" in the menu "Observation window" on the top left of the interface.
8. Select file Book_Fig2_26.irr, click "**cell size**" and **ok** if the cell size appearing in the window "Select a new cell size" is ok and then the small **ok** button in the Select a shape file window. *Programita* now determines the area of the rectangle that belongs to the observation window. Basically, *Programita* generates an underlying grid with a spatial resolution of one bin (i.e., the cell size) and all cells outside are marked and excluded. *Programita* outputs the resulting intensity file as temporary file "int_temp.int".



Observation window ?

○ Analyze all data in rectangle

● Irregularly shaped study region



Define irregularily shaped study area

1) Confirm, or select a new cell size for your data

2) Select the data file providing the boarder of the study region

Note The first data point in your list must lie well in the interior of your study region

Select a shape file

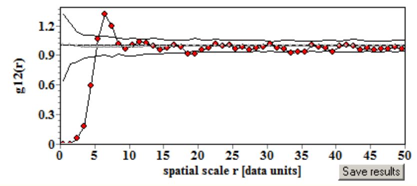
Book_Fig2_26.shp
circle25.shp
patch_Fig2_26.shp

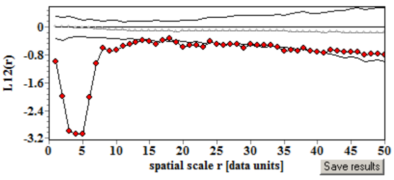
Cell size

Cancel and proceed with all data in plot

9. Click "**Calculate Index**" and *Programita* shows a plot of the data within the reduced observation window. The excluded area is marked in black (see above).
10. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
11. Select "**Pattern 1 fix , pattern 2 CSR**" in the window "Select a null model".
12. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
13. Click the button "**Calculate Index** and *Programita* conducts the analysis. You can observe during the simulations that the null model does indeed not distribute points outside the observation window.

14. The results show for the pair correlation function no departures from the null model at distances larger than 7m (as expected), but small-scale repulsion between pattern 1 and 2 close to the jamming point (as indicated by the higher density of points of pattern 2 around points of pattern 1 just before 6m). In contrast, the shape of the cumulative L-function is more difficult to interpret:

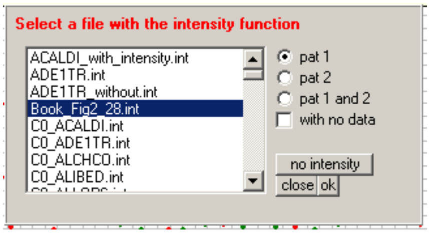




Example InhomBi.res (bivariate pattern, irregular study area)

Now we redo the previous example, but now explicitly based on inhomogeneous second-order summary functions (note that the procedure for irregularly shaped study areas uses them implicitly).

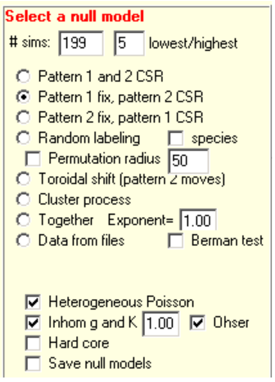
1. Execute *Programita*.
2. Highlight data file InhomBi.dat you want to analyze in Input data and click the small "ok" button.
3. Select a bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button "**Calculate Index**"
7. Click the checkbox "**Calculate simulation envelopes**" to be found in the menu "What do you want to do?" on the top left of the interface.
8. Enable check box "**Inhom g and K**"
9. The window "Select a file with the intensity function" appears where you select the intensity file you want to use for estimation of the inhomogeneous second-order summary functions (i.e., int_Book_Fig2_28.int). Select "**pat 1**" (because it is the intensity of pattern 1) and then click the small "**ok**" button


Select a file with the intensity function

ACALDI_with_intensity.int
ADE1TR.int
ADE1TR_without.int
Book_Fig2_28.int
C0_ACALDI.int
C0_ADE1TR.int
C0_ALCHCO.int
C0_ALIBED.int

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

10. Select "**Pattern 1 fix, pattern 2 CSR**" in the window "Select a null model".


Select a null model

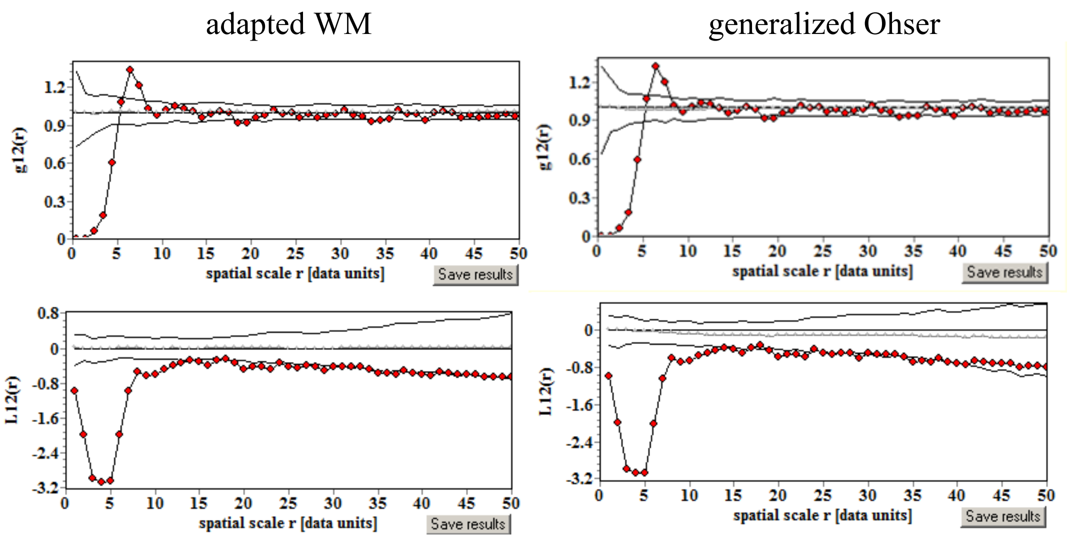
sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
● Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test

☑ Heterogeneous Poisson
☑ Inhom g and K 1.00 ☑ Ohser
□ Hard core
□ Save null models

11. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).Click the button "**Calculate Index** and *Programita* conducts the analysis. You can observe during the simulations that the null model does indeed not distribute points outside the observation window. The results are the same as in the previous example.

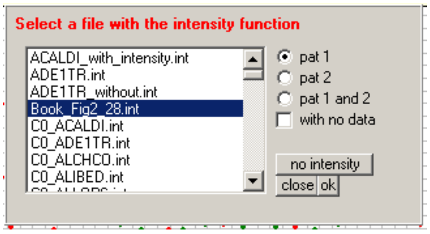
12. The results are identical to the previous example and comparison with the adapted WM estimator shows that the adapted estimator provides better results because it removes the very small bias in the L-function of the null model shown by the generalized Ohser estimator and a somewhat smaller variance for the *L*-function:



Example InhomBi_SoftCore.res

Now we reanalyze the previous example but using the bivariate interaction point process that was used to generate the data.

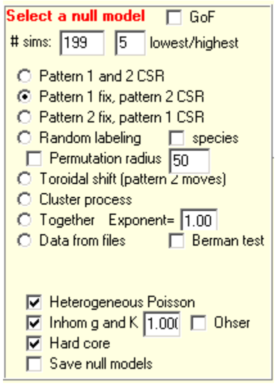
1. Execute *Programita*.
2. Highlight data file InhomBi.dat you want to analyze in Input data and click the small “ok” button.
3. Select a bin of 1m window Select a new cell size
4. Select a ring width of 3 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button “**Calculate Index**”
7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
8. Enable check box “**Inhom g and K**”
9. The window “Select a file with the intensity function” appears where you select the intensity file you want to use for estimation of the inhomogeneous second-order summary functions (i.e., int_Book_Fig2_28.int). Select “**pat 1**” (because it is the intensity of pattern 1) and then click the small “**ok**” button


Select a file with the intensity function

ACALDI_with_intensity.int
ADE1TR.int
ADE1TR_without.int
Book_Fig2_28.int
C0_ACALDI.int
C0_ADE1TR.int
C0_ALCHCO.int
C0_ALIBED.int

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

10. Select “**Pattern 1 fix, pattern 2 CSR**” in the window “Select a null model”.

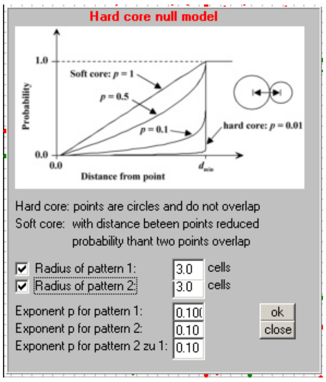

Select a null model □ GoF
# sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
● Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test

☑ Heterogeneous Poisson
☑ Inhom g and K 1.00(□ Ohser
☑ Hard core
□ Save null models

11. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).Click the button "**Calculate Index** and *Programita* conducts the analysis. You can observe during the simulations that the null model does indeed not distribute points outside the observation window. The results are the same as in the previous example.

12. Click checkbox "Hard core" and go to the window "Hard core null model" to define details of the RSA null model. Click "**Radius of pattern 1**" and "**Radius of pattern 2**" because you have a bivariate pattern and provide the radius (3.0) for both patterns in our case and the interaction exponents p = 0.1 for all cases.

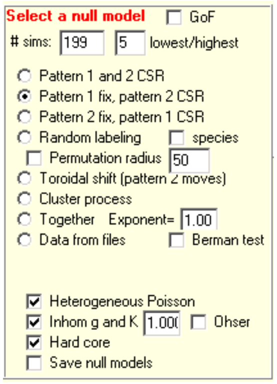


Hard core null model

Hard core: points are circles and do not overlap
Soft core: with distance beteen points reduced
probability thant two points overlap

☑ Radius of pattern 1: 3.0 cells
☑ Radius of pattern 2: 3.0 cells

Exponent p for pattern 1: 0.10(
Exponent p for pattern 2: 0.10
Exponent p for pattern 2 zu 1: 0.10



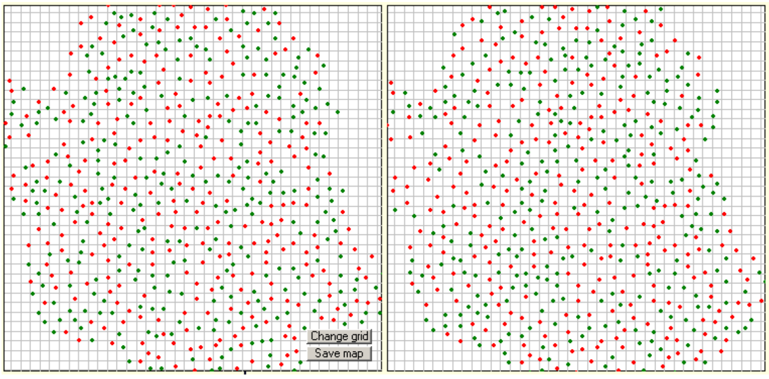
Select a null model □ GoF

# sims: 199 5 lowest/highest

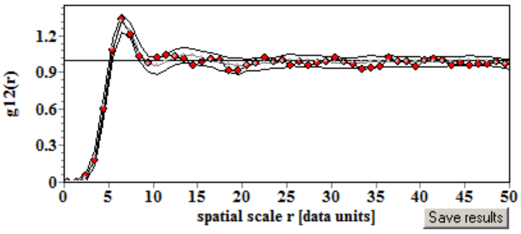
○ Pattern 1 and 2 CSR
● Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test

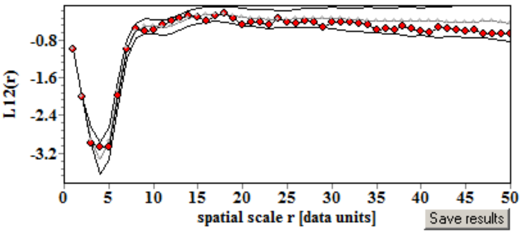
☑ Heterogeneous Poisson
☑ Inhom g and K 1.00(□ Ohser
☑ Hard core
□ Save null models

13. To simulate the point process press "**Calculate Index**". The resulting patterns are regular patterns with repulsion of pattern 1 points, pattern 1 to pattern 2 points and of pattern 2 points:



13. As expected, the pair correlation and L-functions of the data (that were generated with the above parameters of the RSA process) is well within the simulation envelopes:

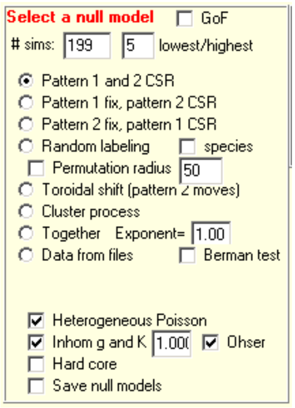




Example Inhom_12HP_Ohser.res

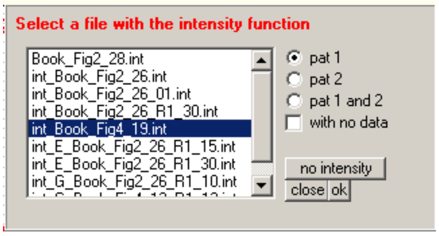
In this example we analyze a data set that was generated with two patterns following both independent heterogeneous Poisson processes based on the intensity function int_Book_Fig4_19.int.

1. Execute *Programita*.
2. Highlight data file Inhom_12HP.dat you want to analyze in Input data and click the small “ok” button.
3. Select a bin of 1m window Select a new cell size
4. Select a ring width of 7 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button “**Calculate Index**”
7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
8. Select “**Pattern 1 and 2 CSR**” in the window “Select a null model”.
9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).



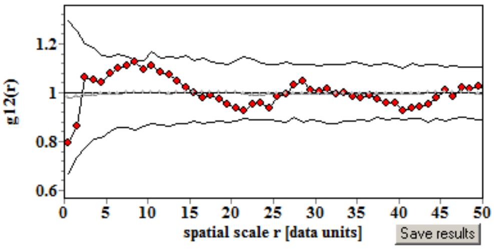
Select a null model □ GoF
# sims: 199 5 lowest/highest
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test
☑ Heterogeneous Poisson
☑ Inhom g and K 1.00(☑ Ohser
□ Hard core
□ Save null models

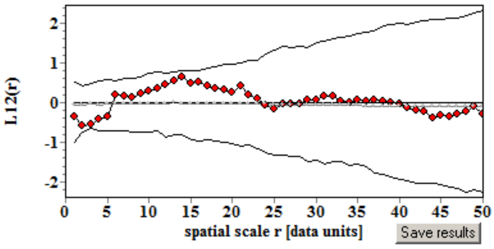
10. Enable check box “**Inhom g and K**” The window “Select a file with the intensity function” appears where you select the intensity file you want to use for estimation of the inhomogeneous second-order summary functions (i.e., int_Book_Fig4_19.int). Select “**pat 1**” (because it is the intensity of pattern 1) and then click the small “**ok**” button.



Select a file with the intensity function
Book_Fig2_28.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_Book_Fig2_26_R1_30.int
int_Book_Fig4_19.int
int_E_Book_Fig2_26_R1_15.int
int_E_Book_Fig2_26_R1_30.int
int_G_Book_Fig2_26_R1_10.int
● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

11. Note that the null model simulates for both patterns independent heterogeneous Poisson process based on the intensity function.
12. Click the button “**Calculate Index** and *Programita* will conduct the analysis.
13. The results show that there are no interactions between the two patterns, once the effect of the shared habitat suitability is removed by the use of inhomogeneous summary functions:



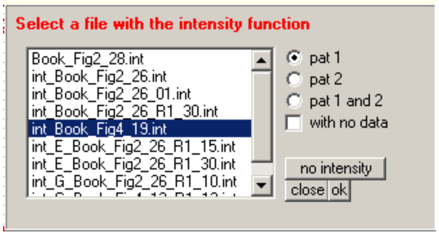


Results file Inhom_2HP_WM.res contains the setting for the case where the null model randomizes only pattern 2 and where the adapted WM estimator is used. **Note that in this case only the intensity function of pattern 2 is needed. This allows you to analyze the interactions of a heterogeneous patterns 2 to an observed pattern 1.**

Example InhomSharedCluster.res

In this example we analyze a data set that was generated with an inhomogeneous Thomas process with shared parents (see also example BiThomasShared.res). again based on the intensity function int_Book_Fig4_19.int. That means that the cluster centers are the same (i.e., shared parents) and the cluster sizes are also the same. The parameters of the original point process were $\sigma_1 = \sigma_2 = 6$ and $\rho_1 = \rho_1$= 0.001. Use a ring width of $dr = 7$ for the analyses.

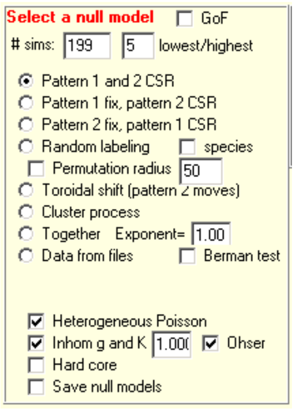
1. Execute *Programita*.
2. Highlight data file inhomShCluster.dat you want to analyze in Input data and click the small “ok” button.
3. Select a bin of 1m window Select a new cell size
4. Select a ring width of 7 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button “**Calculate Index**”
7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
8. Enable check box “**Inhom g and K**” The window “Select a file with the intensity function” appears where you select the intensity file you want to use for estimation of the inhomogeneous second-order summary functions (i.e., int_Book_Fig4_19.int). Select “**pat 1**” (because it is the intensity of pattern 1) and then click the small “**ok**” button.


Select a file with the intensity function

Book_Fig2_28.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_Book_Fig2_26_R1_30.int
int_Book_Fig4_19.int
int_E_Book_Fig2_26_R1_15.int
int_E_Book_Fig2_26_R1_30.int
int_G_Book_Fig2_26_R1_10.int

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

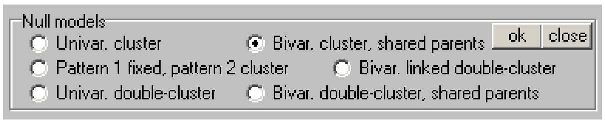
9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
10. Select “**Cluster process**” in the window “Select a null model”.
11. A window “Fit of cluster process to data” opens. Select in the section “Null models” at the bottom the button “**Bivar. cluster, shared parents**”.


Select a null model □ GoF

sims: 199 5 lowest/highest

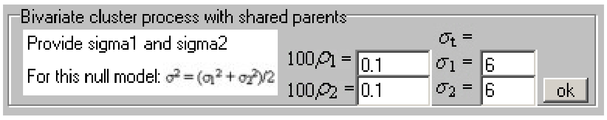
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test

☑ Heterogeneous Poisson
☑ Inhom g and K 1.000 ☑ Ohser
□ Hard core
□ Save null models


Null models

○ Univar. cluster
● Bivar. cluster, shared parents
○ Pattern 1 fixed, pattern 2 cluster
○ Bivar. linked double-cluster
○ Univar. double-cluster
○ Bivar. double-cluster, shared parents

11. Now this window appears that asks you to provide the parameters from the univariate analysis of patterns 1 and 2: $\sigma_1 = \sigma_2 = 6$ and $\rho_1 = \rho_1$= 0.001. Press the small “**ok**” button and again the small “**ok**” button:


Bivariate cluster process with shared parents

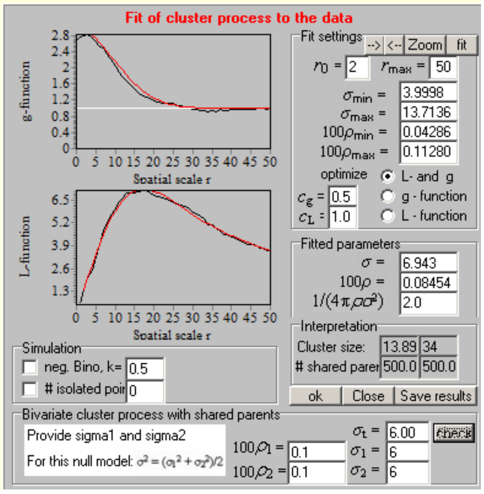
Provide sigma1 and sigma2

For this null model: $\sigma^2 = (\sigma_1^2 + \sigma_2^2)/2$

$100\rho_1$ = 0.1
$100\rho_2$ = 0.1
σ_t =
σ_1 = 6
σ_2 = 6

12. Now you are at the fitting window. Use the "**fit**" and "**zoom**" buttons to find a good fit. *Programita* fits the bivariate inhomogeneous pair correlation function to the data:

$$g_{12}(r,\rho,\sigma_t) = 1 + \frac{1}{\rho}\frac{\exp(-r^2/4\sigma_t^2)}{4\pi\sigma_t^2} \quad \text{where } \sigma_t^2 = (\sigma_1^2+\sigma_2^2)/2$$



Fit of cluster process to the data

Fit settings

r_0 = 2 r_{max} = 50

σ_{min} = 3.9998

σ_{max} = 13.7136

$100\rho_{min}$ = 0.04286

$100\rho_{max}$ = 0.11280

optimize ● L- and g ○ g - function ○ L - function

c_g = 0.5

c_L = 1.0

Fitted parameters

σ = 6.943

100ρ = 0.08454

$1/(4\pi\rho\sigma^2)$ 2.0

Interpretation

Cluster size: 13.89 34

shared parer 500.0 500.0

Simulation

□ neg. Bino, k= 0.5

□ # isolated poir 0

Bivariate cluster process with shared parents

Provide sigma1 and sigma2

For this null model: $\sigma^2 = (\sigma_1^2+\sigma_2^2)/2$

$100\rho_1$ = 0.1

$100\rho_2$ = 0.1

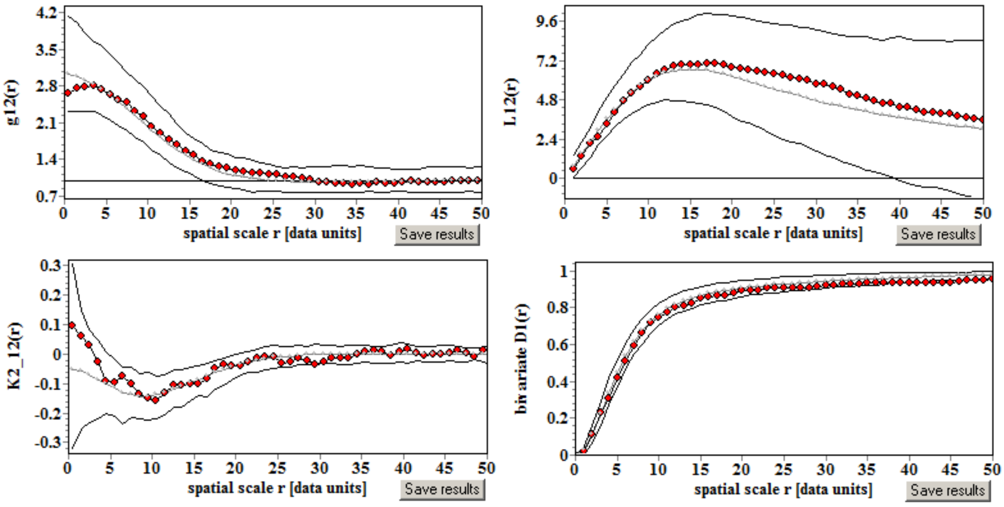
σ_t = 6.00

σ_1 = 6

σ_2 = 6

The best fit yields a parameter *Aρ* = 500 shared parents and the fitted value of σ_t = 6.6 is in good agreement with $\sigma_t = ((\sigma_1^2+\sigma_2^2)/2)^{0.5}$= 6 (click "check" for estimating σ_t. Then click the "ok" button and "Calculate Index".

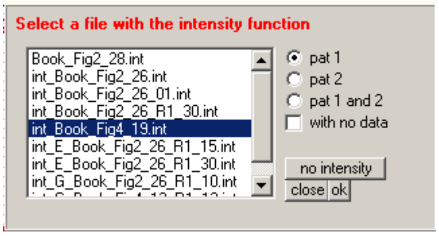
13. As expected, the different summary functions are in good agreement with the fitted point process for both, the univariate analysis of pattern 1 and the bivariate analysis:



Example InhomPartlySharedCluster.res

In this example we analyze a data set that was generated with an inhomogeneous Thomas process with partly shared parents (see also example Book_Fig4_29.res), again based on the intensity function int_Book_Fig4_19.int. That means that part of the cluster centers are the same (i.e., shared parents) and the cluster sizes are also the same. The parameters of the original point process were $\sigma_1 = \sigma_2 = 6$ and $\rho_1 = 0.002$ and $\rho_2 = 0.001$. Use a ring width of $dr = 7$ for the analyses.

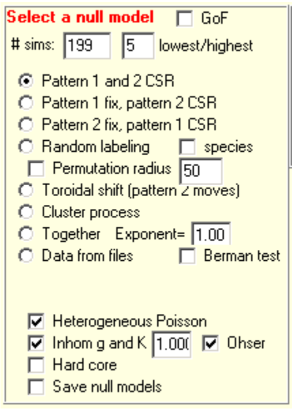
1. Execute *Programita*.
2. Highlight data file inhomPShCluster.dat you want to analyze in Input data and click the small “ok” button.
3. Select a bin of 1m window Select a new cell size
4. Select a ring width of 7 in the menu “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$.
6. Press button “**Calculate Index**”
7. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
8. Enable check box “**Inhom g and K**” The window “Select a file with the intensity function” appears where you select the intensity file you want to use for estimation of the inhomogeneous second-order summary functions (i.e., int_Book_Fig4_19.int). Select “**pat 1**” (because it is the intensity of pattern 1) and then click the small “**ok**” button.

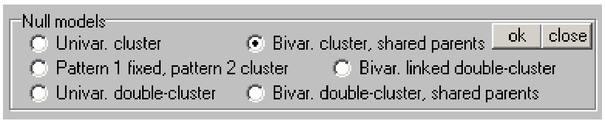

Select a file with the intensity function

Book_Fig2_28.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_Book_Fig2_26_R1_30.int
int_Book_Fig4_19.int
int_E_Book_Fig2_26_R1_15.int
int_E_Book_Fig2_26_R1_30.int
int_G_Book_Fig2_26_R1_10.int

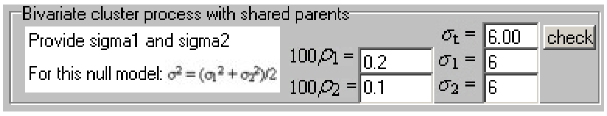
● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

9. Specify the number of simulations of the null model (199 in the example) and the rule for the estimation of simulation envelopes (here the 5th lowest and highest values of the summary function of the 199 simulated null model data sets).
10. Select “**Cluster process**” in the window “Select a null model”.
14. A window “Fit of cluster process to data” opens. Select in the section “Null models” at the bottom the button “**Bivar. cluster, shared parents**”.


Select a null model □ GoF
# sims: 199 5 lowest/highest
● Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
○ Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test
☑ Heterogeneous Poisson
☑ Inhom g and K 1.000 ☑ Ohser
□ Hard core
□ Save null models

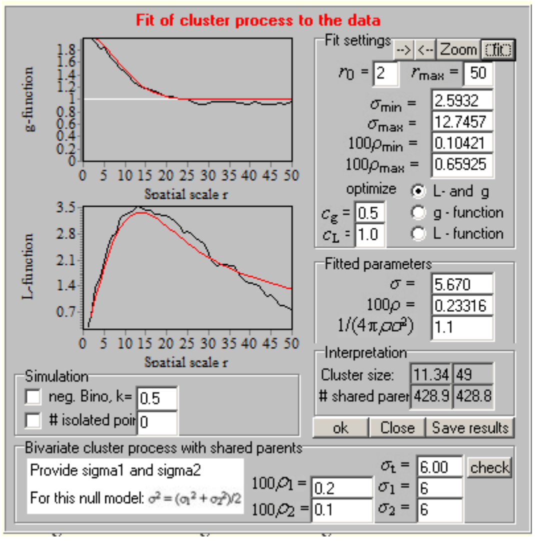

Null models
○ Univar. cluster ● Bivar. cluster, shared parents
○ Pattern 1 fixed, pattern 2 cluster ○ Bivar. linked double-cluster
○ Univar. double-cluster ○ Bivar. double-cluster, shared parents

11. Now this window appears that asks you to provide the parameters from the univariate analysis of patterns 1 and 2: $\sigma_1 = \sigma_2 = 6$, $\rho_1 = 0.002$ and $\rho_2 = 0.002$. Press the small “**ok**” button:


Bivariate cluster process with shared parents
Provide sigma1 and sigma2
For this null model: $\sigma^2 = (\sigma_1^2 + \sigma_2^2)/2$
$100\rho_1$ = 0.2
$100\rho_2$ = 0.1
σ_t = 6.00
σ_1 = 6
σ_2 = 6

15. Now you are at the fitting window. Use the “**fit**” and “**zoom**” buttons to find a good fit. *Programita* fits the bivariate inhomogeneous pair correlation function for the Thpmas process with partly shared parents to the data:

$$g_{12}(r,\rho^*,\sigma_1,\sigma_2) = 1 + \frac{1}{\rho^*} h_2(r,\sigma^*) \text{ where } \sigma^* = \sqrt{(\sigma_1^2+\sigma_2^2)/2} \text{ and } \rho^* = \frac{\rho_1\rho_2}{\rho_s}$$



Fit settings
r_0 = 2 r_{max} = 50
σ_{min} = 2.5932
σ_{max} = 12.7457
$100\rho_{min}$ = 0.10421
$100\rho_{max}$ = 0.65925
optimize ● L- and g ○ g - function ○ L - function
c_g = 0.5
c_L = 1.0

Fitted parameters
σ = 5.670
100ρ = 0.23316
$1/(4\pi\rho\sigma^2)$ 1.1

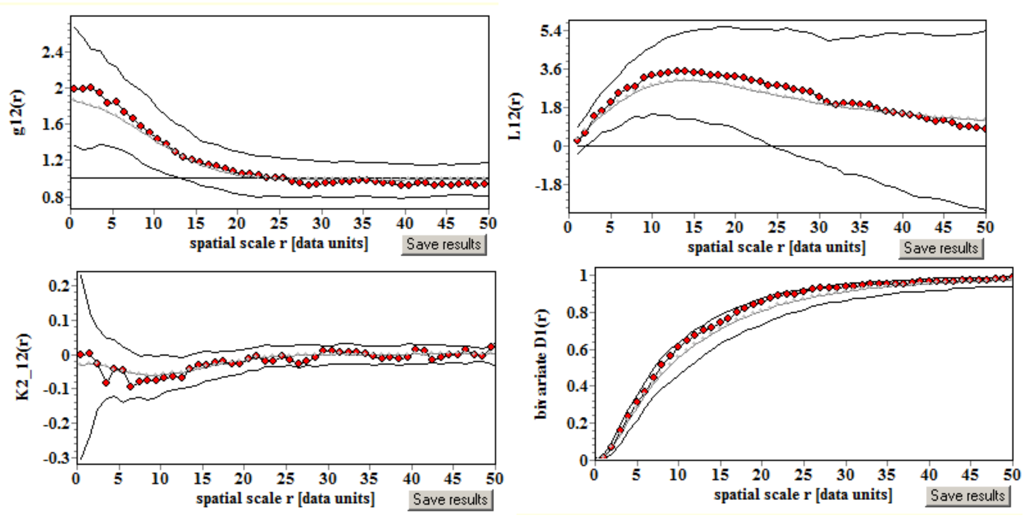
Interpretation
Cluster size: 11.34 49
# shared parer 428.9 428.8

Simulation
□ neg. Bino, k= 0.5
□ # isolated poir 0

Bivariate cluster process with shared parents
Provide sigma1 and sigma2
For this null model: $\sigma^2 = (\sigma_1^2 + \sigma_2^2)/2$
$100\rho_1$ = 0.2
$100\rho_2$ = 0.1
σ_t = 6.00
σ_1 = 6
σ_2 = 6

The best fit yields a parameter $A\rho_s = A(\rho_1\rho_2/\rho^*) = 429$ shared parents (which agrees with the expected number of 500 shared parents), and the fitted value of $\sigma_t = 5.7$ is in good agreement with $\sigma_t = ((\sigma_1^2 + \sigma_2^2)/2)^{0.5} = 6$ (click “check” for estimating σ_t. Then click the “ok” button and “Calculate Index”.

16. As expected, the different summary functions are in good agreement with the fitted point process:



5 Analysis of qualitatively marked patterns

Qualitatively marked patterns usually comprise the coordinates of a univariate pattern, but each point carries an additional (*a posteriori*) qualitative mark that distinguishes two types of points. The qualitative mark is a binary property of the point such as surviving vs. dead, infected vs. non-infected, occupied vs. non-occupied, etc. which was created by a given process *a posteriori* on the existing points of the univariate pattern. This distinguishes (*a posteriori*) qualitative marked patterns from bivariate patterns where the difference between the two types of points is *a priori* (e.g., two different species).

The major interest in analysis of qualitatively marked patterns is in revealing the spatial correlation structure of the marking process, conditional on the given univariate pattern. For example, are dead saplings clustered within all saplings, are dead saplings surrounded by a higher density of saplings than surviving saplings, etc.? The basic null model for this data type is the so-called **random labeling null model** which randomly shuffles the marks over the points, thus removing all potential spatial structure in the marks. Chapter 4.4 in Wiegand and Moloney (2014) provides examples for the different analyses of qualitatively marked patterns that are useful in ecology.

5.1 Getting started

5.1.1 Data preparation

Qualitatively marked patterns comprise the coordinates of the underlying univariate point patterns and the mark. The data files for standard analysis must be an ASCII file with the *.dat extension and the following format (the example shows the first lines of the file Book_Fig_2_15.dat):

```
0  500  0  500  600
0.60   35.35  0  1
0.70  274.90  0  1
1.10  385.85  1  0
1.15  342.20  1  0
1.80  274.60  1  0
2.02  385.30  0  1
2.50  230.25  1  0
2.60  383.05  1  0
2.85   40.15  0  1
3.25  322.25  0  1
3.65   37.45  0  1
…
```

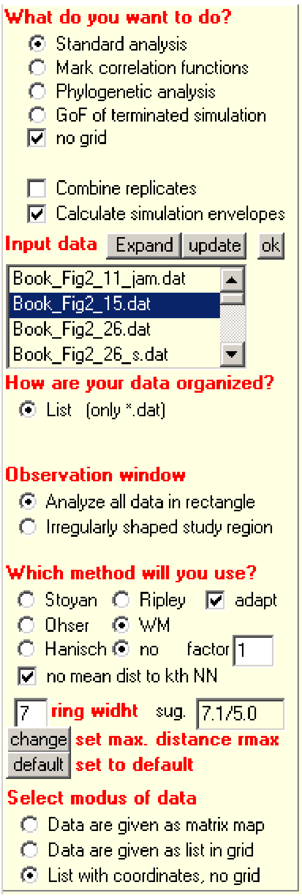
where the first line gives the size of the observation window (500 × 500 units in the example) and the number of points in the underlying univariate pattern (= number of lines following the header). The first two columns are the coordinates, an entry “1” in the third column indicates that the point is of type 1 (i.e., dead in the example) and an entry “1” in the fourth column indicates that the point is of type 2 (i.e., surviving in the example). The value of the third and the forth columns must be for the standard analysis mode “0 1” or “1   0”, no larger numbers or “1 1” are allowed.

The data file must be a space or tab delimited ASCII file with the *.dat extension. If you use Excel, there is a simple, but obviously generally unknown, way of saving files of a given type with a given extension:

1. Prepare the data file in Excel following the instructions above.
2. Then save as a tab delimited text file, but write "name.dat" for the name (usually you would only write name and end up with a file named name.txt. The quotation marks are important because they force Excel to save the comma delimited file under the name name.dat.

5.1.2 Steps of random labeling analysis in standard mode

Programita estimates for data files of this type several adapted test statistics based on pair correlation functions (or *L*- and *K*-functions) presented initially in Jacquemyn et al. (2010) and detailed in section 4.4.1 in Wiegand and Moloney (2014). The standard analysis mode can be accessed with the following sequence of actions:



What do you want to do?

● Standard analysis

○ Mark correlation functions

○ Phylogenetic analysis

○ GoF of terminated simulation

☑ no grid

□ Combine replicates

☑ Calculate simulation envelopes

Input data Expand update ok

Book_Fig2_11_jam.dat

Book_Fig2_15.dat

Book_Fig2_26.dat

Book_Fig2_26_s.dat

How are your data organized?

● List (only *.dat)

Observation window

● Analyze all data in rectangle

○ Irregularly shaped study region

Which method will you use?

○ Stoyan ○ Ripley ☑ adapt

○ Ohser ● WM

○ Hanisch ● no factor 1

☑ no mean dist to kth NN

7 ring widht sug. 7.1/5.0

change set max. distance rmax

default set to default

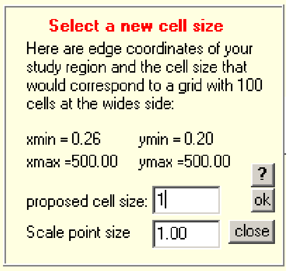
Select modus of data

○ Data are given as matrix map

○ Data are given as list in grid

● List with coordinates, no grid

1. Select "**Standard analysis**" in window What do you want to do?
2. Highlight a data file in Input data ("Book_2_15.dat" in the example).
3. The window Select a new cell size opens and allows you to provide a bin for your analysis given in units of your data. For example, if your data are in meter units and your observation window is 500 × 500m in size, an appropriate bin would be 1m. Press "**ok**" to confirm selection of the bin.



Select a new cell size

Here are edge coordinates of your study region and the cell size that would correspond to a grid with 100 cells at the wides side:

xmin = 0.26 ymin = 0.20

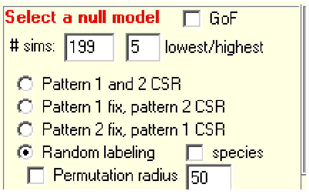
xmax =500.00 ymax =500.00

?

proposed cell size: 1 ok

Scale point size 1.00 close

4. Select a ring width of 7 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$ (will not be used in random labeling analyses).
6. To access the standard analysis mode for **qualitatively marked patterns** (i.e., random labeling analysis; **data type 4**) you must enable the checkbox "Calculate simulation envelopes" in the window What do you want to do? and select the random labeling null model in the Select a null model window.



Select a null model □ GoF

sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR

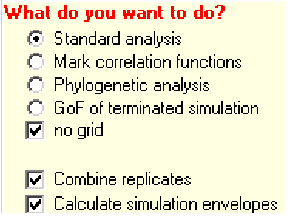
○ Pattern 1 fix, pattern 2 CSR

○ Pattern 2 fix, pattern 1 CSR

● Random labeling □ species

□ Permutation radius 50

7. If you enable the check box “**Combine replicates**” before running the analysis and save the results of the analysis with Save results under name.res, *Programita* saves additionally two files (name_1.rep and name_2.rep) which allow you to view and save the results for all different test statistics. You can access the procedure for loading the results with button “**Replicates**”.

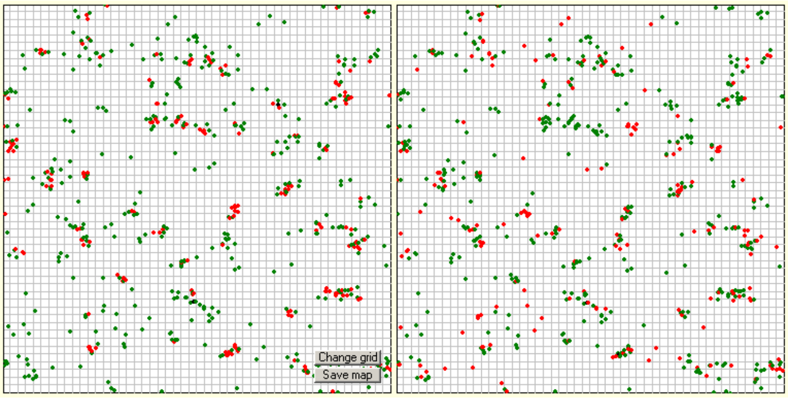


What do you want to do?

● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

☑ Combine replicates
☑ Calculate simulation envelopes

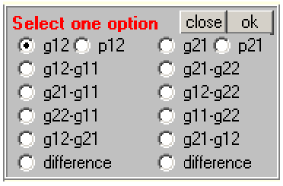
8. Press button “**Calculate Index**” and *Programita* runs the desired number of simulations of the random labeling null model:



As you can see, the randomization procedure keeps the locations of the points unchanged, but randomly shuffles the mark, indicated here by red (dead) and green (surviving).

9. After running the Monte Carlo simulations of the random labeling null model you can select among several test statistics that allow you to assess potential departures from random labeling. The test statistics are described in detail in Jacquemyn et al. (2010) and section 4.4 on Wiegand and Moloney (2014). Here the ones on the left (the ones on the right results from exchanging labels 1 and 2):

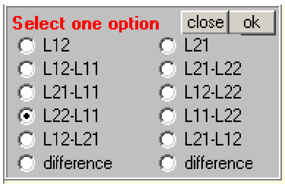
   - $g_{12}(r)$: shows the pair correlation functions $g_{11}(r)$ and $g_{12}(r)$
   - $p_{12}(r)$: shows the mark connection functions $p_{11}(r)$ and $p_{12}(r)$
   - $g_{12}(r)$ - $g_{11}(r)$ shows the $g_{11}(r)$ and $g_{12}(r)$ - $g_{11}(r)$
   - $g_{21}(r)$ - $g_{11}(r)$ shows the $g_{11}(r)$ and $g_{21}(r)$ - $g_{11}(r)$
   - $g_{22}(r)$ - $g_{11}(r)$ shows the $g_{11}(r)$ and $g_{22}(r)$ - $g_{11}(r)$
   - $g_{12}(r)$ - $g_{21}(r)$ shows the $g_{11}(r)$ and $g_{12}(r)$ - $g_{21}(r)$
   - difference shows the $g_{11}(r)$ and $g_{1,1+2}(r)$ - $g_{2,1+2}(r)$



Select one option close ok

● g12 ○ p12 ○ g21 ○ p21
○ g12-g11 ○ g21-g22
○ g21-g11 ○ g12-g22
○ g22-g11 ○ g11-g22
○ g12-g21 ○ g21-g12
○ difference ○ difference

10. If you select the *L*-function before selecting the random labeling null model you can also view the analogous test statistic based on the *L*- or *K*-functions.

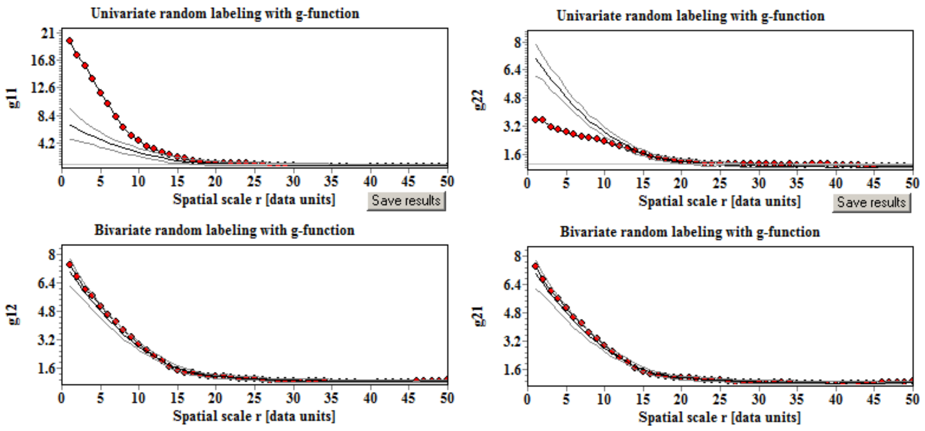


Select one option close ok

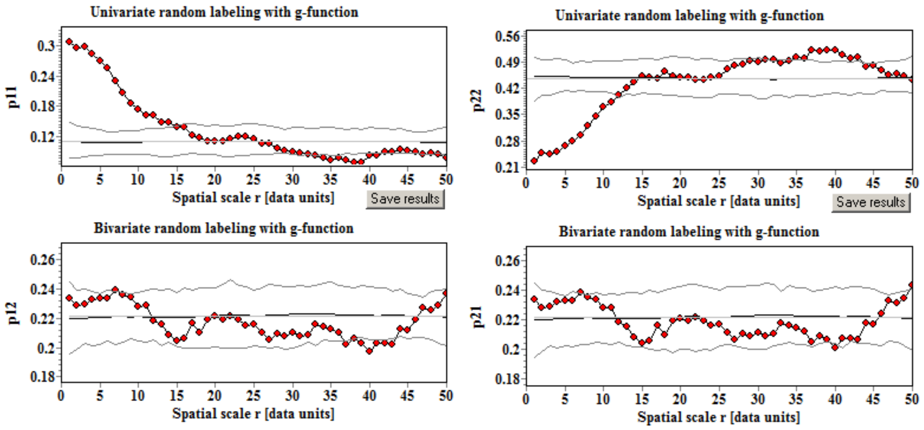
○ L12 ○ L21
○ L12-L11 ○ L21-L22
○ L21-L11 ○ L12-L22
● L22-L11 ○ L11-L22
○ L12-L21 ○ L21-L12
○ difference ○ difference

15. The marks of the 600 points of data Book_Fig2_15.dat are constructed; 200 dead individuals and 400 surviving individuals. The probability of an individuals dying was proportional to the number of individuals occurring within a 10 m neighborhood; more isolated individuals thus had a higher probability of surviving.

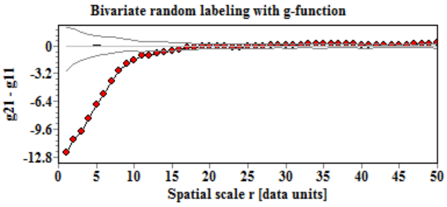
16. The results of the univariate and bivariate pair correlation functions show that the dead individuals are strongly clustered inside all individuals (g11), that the surviving individuals are regularly distributed within all individuals (g22), but that the bivariate relationship between surviving and dead (g12, g21) does not show departures from random labeling:

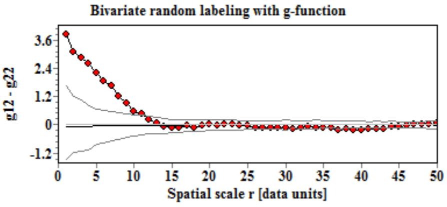


17. When removing the signal of the clustering of all individuals (i.e., the pattern of the underlying univariate pattern) by using the corresponding mark connection functions we can see these results much clearer:

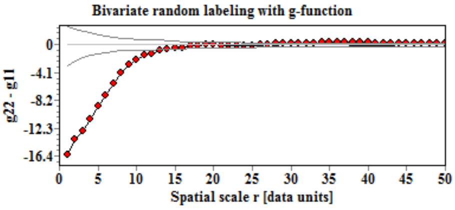


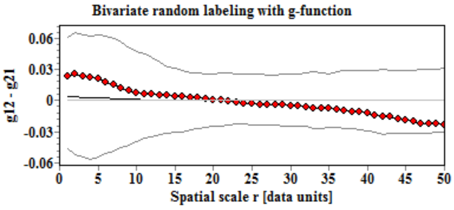
18. Instructive are also the summary functions $g_{21}(r) - g_{11}(r)$ and $g_{12}(r) - g_{22}(r)$. The $g_{21}(r) - g_{11}(r)$ shows that the neighborhood density of dead individuals is much lower around surviving individuals (g21) than dead individuals (g11). The $g_{12}(r) - g_{22}(r)$ shows that the neighborhood density of surviving individuals is somewhat higher around dead individuals (g12) than surviving individuals (g22):





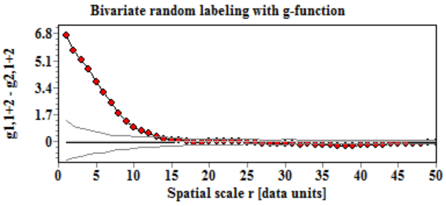
19. The summary functions $g_{22}(r) - g_{11}(r)$ shows that the dead individuals are much more clustered than the surviving ones





and the summary functions $g_{12}(r) - g_{21}(r)$ shows that edge effects are unimportant. Note that the $g_{12}(r)$ and $g_{21}(r)$ are the same except small edge effects that arise for example if many dead individuals are close to an edge of the observation window, but not surviving individuals.

20. Finally, the summary function $g_{1,1+2}(r) - g_{2,1+2}(r)$ which was especially designed to detect density dependent effects in mortality shows that the neighborhood density of surviving and dead individuals (indicated by subscript 1+2) is much higher in neighborhoods around dead individuals than around surviving individuals. Thus, the more pre-mortality individuals in the neighborhood of an individual, the higher the risk of mortality. As expected this effect is strong up to 10m.



Velázquez et al. (2016a) used the different random labeling test functions to analyze the mortality of samplings at the BCI forest.

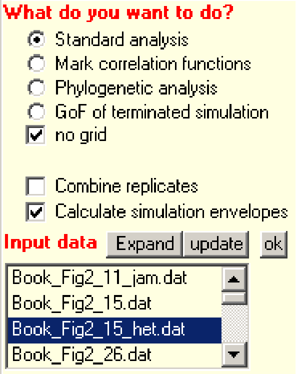
5.2 More complex null models

5.2.1 Local random labeling

Random labeling analysis may also be impacted by heterogeneity. In this case the value of the marks may be influenced by environmental covariates and we may observe systematic spatial trends in the marks. For example, the proportion of dead individuals may be larger at the eastern part of an observation window than at the western part.

The effect of a large-scale heterogeneity in the marking can be factored out in a similar way as using the heterogeneous Poisson process with kernel intensity estimate for factoring out large scale heterogeneity in univariate patterns. While the marks in standard random labeling are shuffled in a way that the mark of each point can be exchanged with the mark of each other point in the entire observation window, localized random labeling exchanges only marks of points which are located closer than a given distance *R*. This removes any small-scale correlation structure of the marks, but maintains their observed large-scale correlation structure. Technically, all n_{1+2} points i of the marked pattern are numbered and the entries of the array *nr*[i] that runs from 1 to n_{1+2} are randomly permutated in a way that the coordinates of all points i and j = *nr*[i] are not farther away than distance *R*.

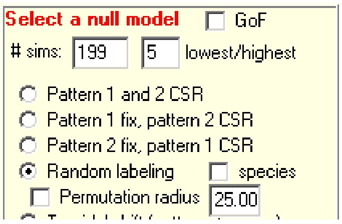
Local random labeling, example Book_Fig2_15_local.res


What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid
☐ Combine replicates
☑ Calculate simulation envelopes
Input data

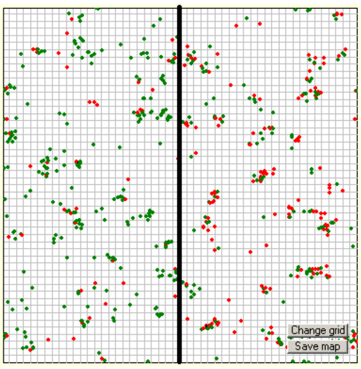
Book_Fig2_11_jam.dat
Book_Fig2_15.dat
Book_Fig2_15_het.dat
Book_Fig2_26.dat

The example pattern Book_Fig2_15_het.dat is the same univariate pattern as used in the previous example, but the marking is different. The points in the left part of the observation window (x-coordinate <250) have a mortality rate of 0.08 and the points of the right part of the observation window (x-coordinate > 250) have a mortality rate of 0.2533.

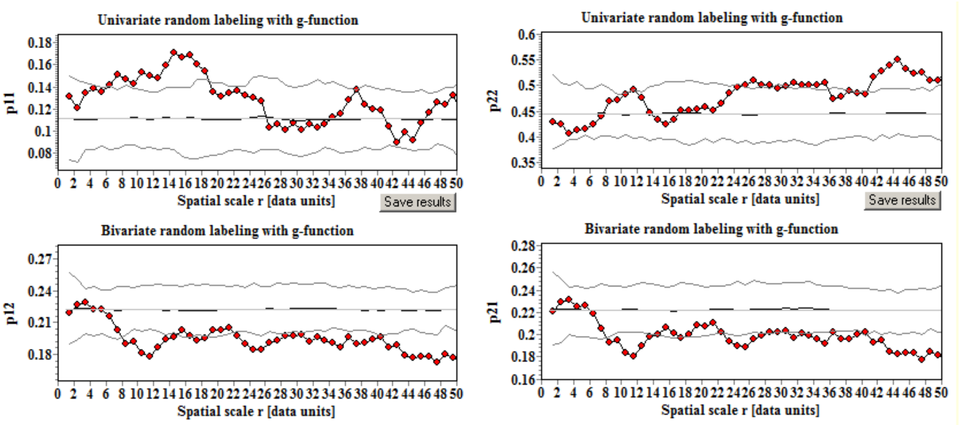
1. Select "**Standard analysis**" in window What do you want to do?
2. Highlight a data file in Input data ("Book_2_15het.dat" in the example).
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 5 in the menu "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$ (will not be used in random labeling analyses).
6. To access the standard analysis mode for **qualitatively marked patterns** (i.e., random labeling analysis; **data type 4**) you must enable the checkbox "Calculate simulation envelopes" in the window What do you want to do? and select the random labeling null model in the Select a null model window.
7. Press button "**Calculate Index**" and *Programita* runs the desired number of simulations of the random labeling null model (e.g., 199).


Select a null model ☐ GoF
# sims: 199 5 lowest/highest
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
● Random labeling ☐ species
☐ Permutation radius 25.00

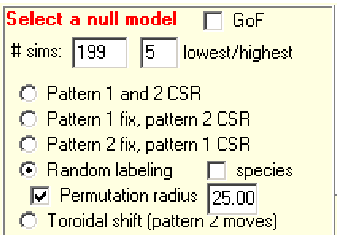
8. You can observe the lower mortality rate at the left side of the observation window:



9. Consequently, the test statistics show departures from random labeling which are difficult to interpret:



10. To conduct local random labeling click the checkbox “**Permutation radius**” and insert an appropriate radius *R* (25m in our case). Only the marks of points that are less than 25m apart are shuffled. This null model therefore maintains approximately the large-scale structure of the marks (i.e., the higher mortality rate on the right side of the observation window).



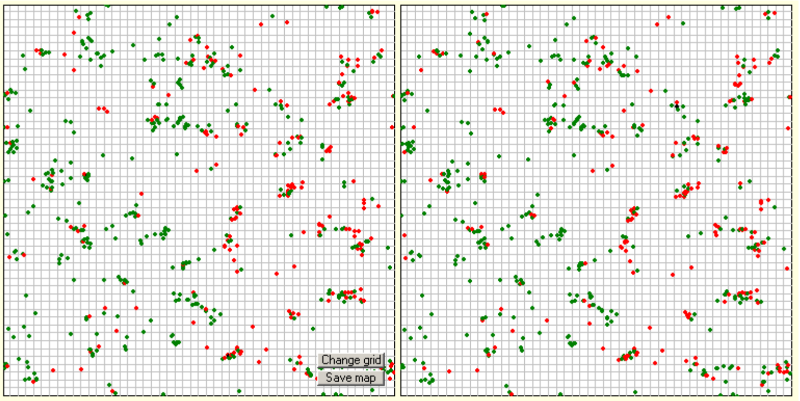
Select a null model ☐ GoF

sims: 199 5 lowest/highest

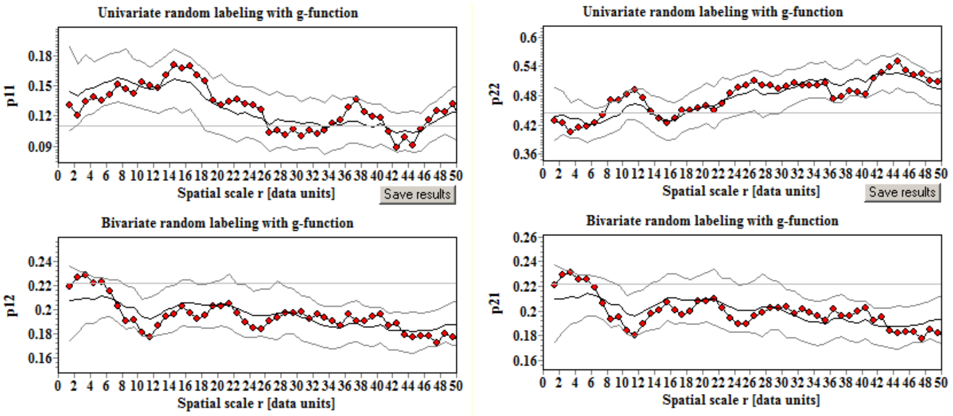
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
● Random labeling ☐ species
☑ Permutation radius 25.00
○ Toroidal shift (pattern 2 moves)

11. Press button “**Calculate Index**” and *Programita* runs the simulations of the random labeling null model:

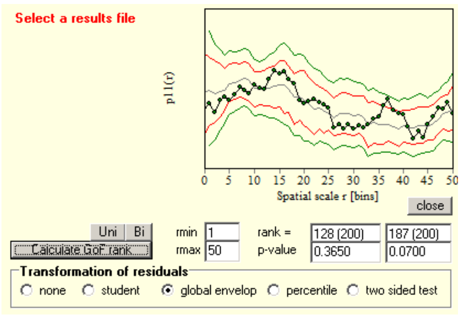
You can observe that the randomization now maintains the large-scale structure of the marks (i.e., the higher mortality rate on the right side of the observation window):



12. The summary functions do now not show significant departures from the local random labeling null model, but the expectation under this null model are not centered on the expectations for random labeling (which are indicated by a grey horizontal line):

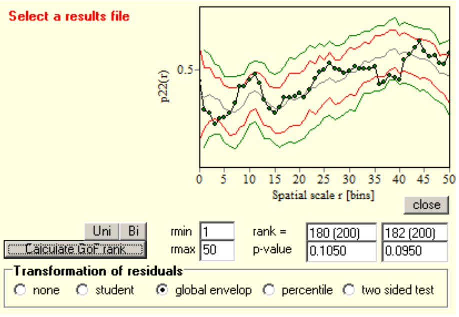


You can verify with the GoF test that there is no significant departure from the null model over distance interval 1-50m:



rmin 1
rmax 50
rank = 128 (200) 187 (200)
p-value 0.3650 0.0700

Transformation of residuals
○ none ○ student ● global envelop ○ percentile ○ two sided test



rmin 1
rmax 50
rank = 180 (200) 182 (200)
p-value 0.1050 0.0950

Transformation of residuals
○ none ○ student ● global envelop ○ percentile ○ two sided test

13. It is clear that the radius *R* should not be too large compared with the scale of the heterogeneity. However, isolated points (with distances to the nearest neighbor < *R*) maintain under this null model their mark.

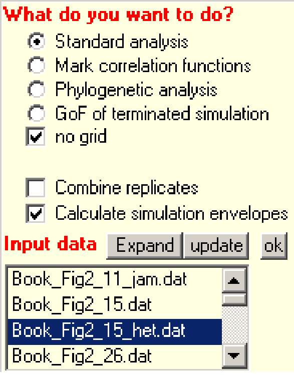
5.2.2 Random labeling with covariate

Random labeling analysis may also be impacted by heterogeneity. In this case the value of the mark may be influenced by environmental covariates and we may observe systematic trends in the marks. For example, the proportion of dead individuals may be larger at the eastern part of an observation window than at the western part.

A second option to consider such first-order heterogeneity in the marks is to use a covariate that describes how the probability of mortality changes in dependence on the location **x**. *Programita* offers you the possibility to read an intensity function $\lambda(\mathbf{x})$ that governs the probability of occurrence of one of the two types of points (e.g., the probability of mortality).

Random labeling with covariate, example Book_Fig_2_15_cov.res

We use the pattern Book_Fig_2_15_het.dat as example. The points in the left part of the observation window (x-coordinate <250) have a mortality rate of 0.08 and the points of the right part of the observation window (x-coordinate > 250) have a mortality rate of 0.25.



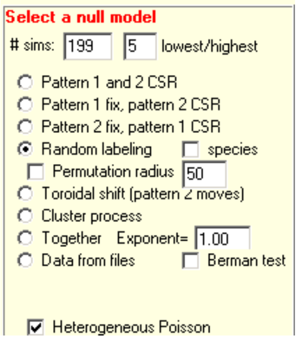
What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ no grid

□ Combine replicates
☑ Calculate simulation envelopes

Input data

Book_Fig2_11_jam.dat
Book_Fig2_15.dat
Book_Fig2_15_het.dat
Book_Fig2_26.dat

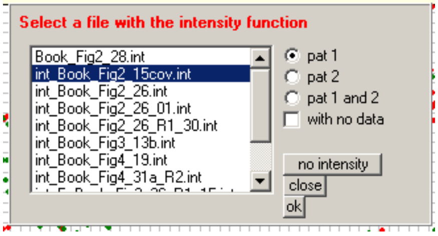
1. Select “**Standard analysis**” in window What do you want to do?
2. Highlight a data file in Input data (“Book_2_15het.dat” in the example) and click the small “ok” button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 5m in window “Which method will you use”
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$ (will not be used in random labeling analyses).
6. To access the standard analysis mode for **qualitatively marked patterns** (i.e., random labeling analysis; **data type 4**) you must enable the checkbox “**Calculate simulation envelopes**” in the window What do you want to do? and select the “Random labeling” in the Select a null model window.



Select a null model
# sims: 199 5 lowest/highest
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
● Random labeling □ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files □ Berman test

☑ Heterogeneous Poisson

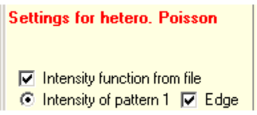
7. To use the covariate press button “**Heterogeneous Poisson**” in the Select a null model window, enable the checkbox “Intensity function from file” in the Settings for hetero. Poisson window and then highlight the intensity file in the corresponding window (int_Book_Fog2_15cov.int in the example).



Select a file with the intensity function

Book_Fig2_28.int
int_Book_Fig2_15cov.int
int_Book_Fig2_26.int
int_Book_Fig2_26_01.int
int_Book_Fig2_26_R1_30.int
int_Book_Fig3_13b.int
int_Book_Fig4_19.int
int_Book_Fig4_31a_R2.int

● pat 1
○ pat 2
○ pat 1 and 2
□ with no data

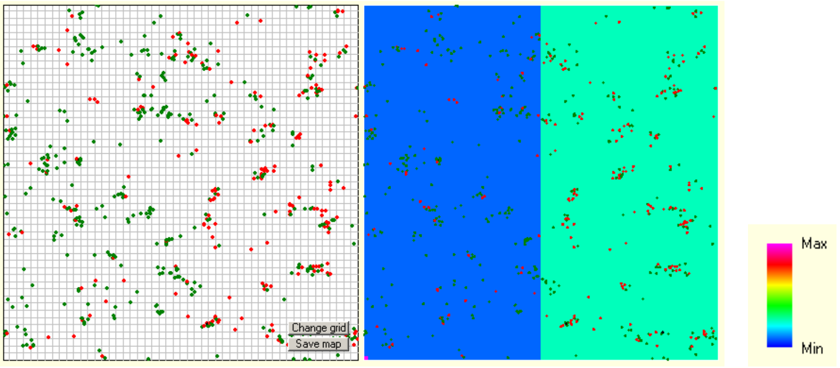


Settings for hetero. Poisson

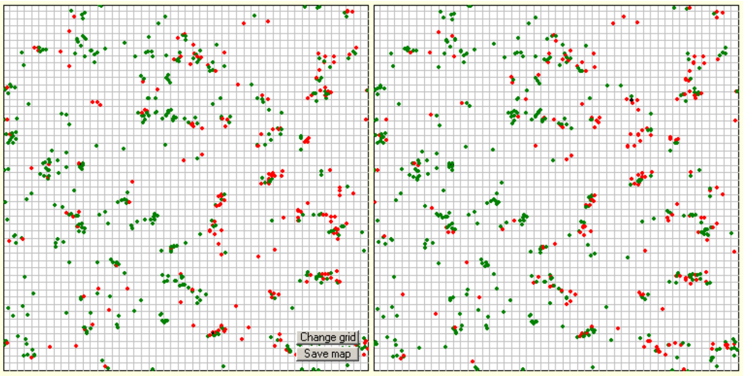
☑ Intensity function from file
● Intensity of pattern 1 ☑ Edge

8. Select “**pat 1**” because the dead individuals are pattern 1 (i.e., the type 1 points). If you select “pat2” the event will be surviving and the probability of the intensity function will be applied to survival and not to mortality.

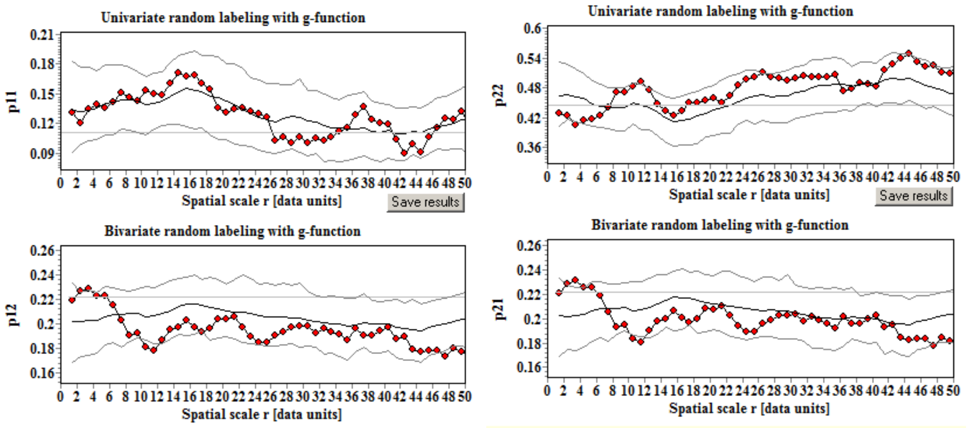
9. *Programita* then shows the intensity function together with the points. The intensity function yields the probability that the event of pattern 1 (here dead) occurs (0.08 at the western part and 0.2533 on the eastern part of the observation window):



10. Now click "**Calculate Index**" to run the simulations of the heterogeneous random labeling null model. You can observe that the randomization maintains the large-scale structure of the marks (i.e., the higher mortality rate on the right side of the observation window):



11. Consequently, the test statistics show no departure from random labeling and similarly to localized random labeling, and the expectation under the null model (black line) differs from that under random labeling (grey horizontal line):



5.2.3 Trivariate random labeling

The standard random labeling explores the statistical properties of the marking process. In many cases, the marking of a given pattern may be dependent on the presence of a third, antecedent pattern. For example, mortality of saplings in tropical forests may depend on the proximity of large trees. Trivariate random labeling explores the influence of an antecedent pattern on the marking of a qualitatively marked pattern.

I included trivariate random labeling into the 2004 *Programita* manual using the name "Random labeling under antecedent condition". Trivariate random labeling based on the *K*-function was first published by Marcelino de la Cruz in Spanish [De la Cruz Rot M. 2006], later in De la Cruz et al. (2008) using the name "independent labeling" and the *Programita* implementations based on the mark connection functions were published in Biganzoli et al. (2009), Jacquemyn et al. (2010) and Raventos et al. (2010). A publication using the grid-based implementation of trivariate random labeling is Murphy et al. (2017).

The null model is again random labeling, but the summary function considers the impact of individuals of the third antecedent pattern on the marking of the qualitatively marked pattern. In this case, we have individuals of the antecedent pattern (subscript a), dead individuals (subscript 1) and surviving individuals (subscript 2) of the marked pattern. The summary function estimates the probability of mortality (or survival) of the individuals of the marked pattern as a function of distance *r* from the individuals of the antecedent pattern:

$$p_{a,1}(r) = \frac{\lambda_1}{(\lambda_1 + \lambda_2)} \frac{g_{a,1}(r)}{g_{a,1+2}(r)}$$

The quantities λ_1 and λ_2 are the partial intensities of the dead and surviving individuals, respectively, and $g_{\mathrm{a},1+2}(r)$ and $g_{\mathrm{a},1}(r)$ are the bivariate pair correlation functions measuring the intensity normalized neighborhood density around antecedent individuals (a) of the combined pattern of surviving and dead individuals (1+2) and dead individuals (1), respectively. More details on trivariate random labeling are provided in sections 3.1.6.2 "Trivariate Random Labeling (Data Type 5)" and 4.4.1.6 "Trivariate Perspective" in Wiegand and Moloney (2014). Note that the summary function used for trivariate random labeling can be considered a mark connection function (equation 3.82) in Wiegand and Moloney (2014) or a mark correlation function (equation 3.83) in Wiegand and Moloney (2014).

Programita offers two equivalent implementations of trivariate random labeling, one based on the grid-based mode and one based on the mark correlation function.

5.2.4 Grid-based trivariate random labeling

You can run the trivariate random labeling using the grid-based implementation of *Programita*. The continuous summary function can be very well approximated with an underlying grid for distances larger than the grid size. The coordinates are given here as coordinates of a grid that runs from 1 to 1000:

```
  1  1000  1  1000  1001
543   952  1  0
...
383   333  1  0
  0     0  1  0
  2   550  0  0
…
993   766  0  0
491   235  0  1
  3   772  0  1
…
```

where the first line gives the size of the observation window (1000 × 1000 units in the example) and the total number of points (= number of lines following the header). The points of the antecedent pattern are coded as

```
543   952  1  0
```

with the coordinates (giving the number of the grid cell in x- and y-direction) and as “1 0” in the third and forth column, respectively.

The points of the event (e.g., dead) in the qualitatively marked pattern are codes as:

```
491   235  0  1
```

with the coordinates (giving the number of the grid cell in x- and y-direction) and as “0 1” in the third and forth column, respectively.

The points of the no-event (e.g., surviving) in the qualitatively marked pattern are codes as:

```
  2   550  0  0
```

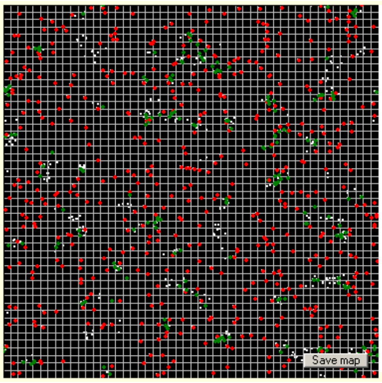
with the coordinates (giving the number of the grid cell in x- and y-direction) and as “0 0” in the third and forth column, respectively.

If there are two of more individuals of the antecedent or qualitatively marked pattern within one cell, each individual must appear individually in the list. *Programita* then reads the list of individuals and applies random labeling over the qualitatively marked pattern.

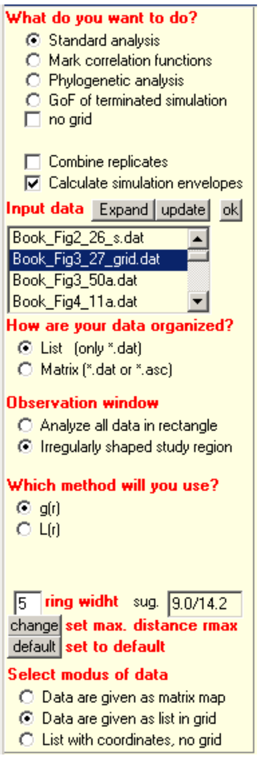
In the grid-based version of *Programita* I managed to handle 3 patterns at the same time with a trick based on irregularly shaped observation windows. The observation window was reduced to cells that contained points, and the random labeling was conducted between cells coded with “0 1” and “0 0”. With small cell sizes the continuos functions were well approximated.

Grid-based trivariate random labeling. Example *Book_Fig3_27_grid.res*

1. Execute *Programita*.
2. Highlight data file Book_Fig3_27_grid.dat you want to analyze in Input data.
3. Disable the option “no grid”
4. Select “**Data are given as list in grid**” in Select modus of data
5. Select a ring width of 5 in the menu “Which method will you use”
6. Click button “change” below to set maximal distance *r* to be analyzed. Insert 100 in small box that opens and then the small **ok** button.
7. Press button “**Calculate Index**”
8. Select “**Irregularly shaped study region**” in window Observation window
9. Press button “**Calculate Index**”. *Programita* now shows a somewhat unusual plot of the data:



where the points of the antecedent pattern are shown in red, the “event” of the qualitatively marked pattern is shown in green, and the no-event in white. All cells which have no point are black. Thus, basically, the observation window is reduced to the cells which contain points.



What do you want to do?
● Standard analysis
○ Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
□ no grid

□ Combine replicates
☑ Calculate simulation envelopes

Input data

Book_Fig2_26_s.dat
Book_Fig3_27_grid.dat
Book_Fig3_50a.dat
Book_Fig4_11a.dat

How are your data organized?
● List (only *.dat)
○ Matrix (*.dat or *.asc)

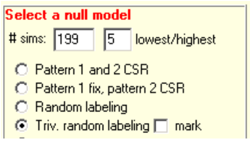
Observation window
○ Analyze all data in rectangle
● Irregularly shaped study region

Which method will you use?
● g(r)
○ L(r)

5 ring widht sug. 9.0/14.2
change set max. distance rmax
default set to default

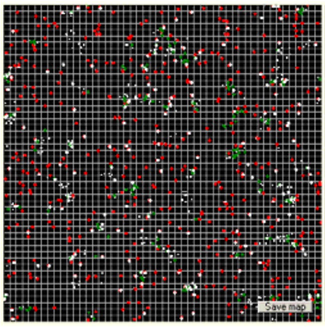
Select modus of data
○ Data are given as matrix map
● Data are given as list in grid
○ List with coordinates, no grid

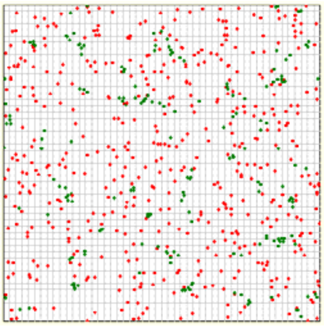
10. Click the checkbox “**Calculate simulation envelopes**” to be found in the menu “What do you want to do?” on the top left of the interface.
11. Select null model “Triv. random labeling” and provide the desired number of simulations of the null model (199 in the example) and the rule for the simulation envelopes (5 in the example).



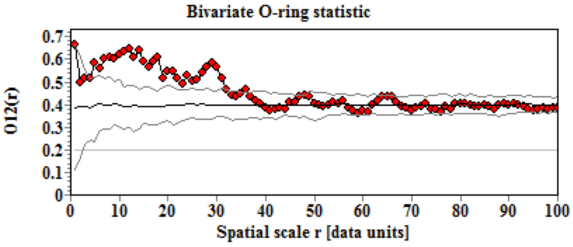
Select a null model
# sims: 199 5 lowest/highest
○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Random labeling
● Triv. random labeling □ mark

12. Press button “**Calculate Index**”. *Programita* now conducts the random labeling of the qualitatively marked pattern, shown are the re-locations of the event:





13. *Programita* now shows the results of the trivariate random labeling analysis:



The label mortality is not random. The antecedent pattern indeed increases the risk of mortality within 40m distance of the points of the antecedent pattern.

5.2.5 Trivariate random labeling based on mark correlation functions

The summary function of trivariate random labeling can be interpreted as a bivariate mark correlation function (equation 3.83 in Wiegand and Moloney (2014):

$$\hat{p}_{al}(r) = \frac{\hat{\rho}_{al}(r)}{\hat{\rho}_{a,l+m}(r)} = \frac{\sum_{i=1}^{n_a}\sum_{j=1}^{n_{l+m}} \mathbf{C}_l(\mathbf{x}_j)\times k(\|\mathbf{x}_i-\mathbf{x}_j\|-r)}{\sum_{i=1}^{n_a}\sum_{j=1}^{n_{l+m}} k(\|\mathbf{x}_i-\mathbf{x}_j\|-r)}$$

where the $\mathbf{x}_\mathrm{i}$ are the points of the antecedent pattern *a* and the $\mathbf{x}_\mathrm{j}$ are the points of the qualitatively marked pattern. The estimator basically visits all pairs of points $\mathbf{x}_\mathrm{i}$ of the antecedent pattern and $\mathbf{x}_\mathrm{j}$ of the qualitatively marked pattern which are located at distance *r* [(selected by the kernel function *k*()] and estimates the mean value of the test function $\mathbf{C}_\mathrm{l}(\mathbf{x}_\mathrm{j})$ over these point pairs. The test function is only a function of the mark of the qualitatively marked pattern and yields 1 if the event occurred for point $\mathbf{x}_\mathrm{j}$ and zero otherwise. If the event is dead, this summary function thus estimates the mean proportion of dead individuals of the qualitatively marked pattern at distance *r* of the individuals of the antecedent pattern *a*. This is a so-called *r*-mark correlation function.

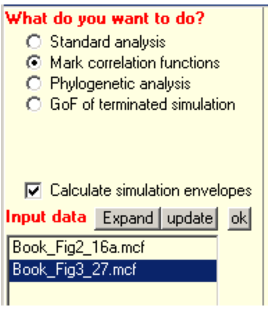
The coding of the data file follows that of bivariate mark correlation functions. It is **data type 9** (a bivariate pattern with one quantitative mark), the data files must be an ASCII file with the *. mcf extension and have always the following format:

```
0  1000   0  1000  1001
543  952   1  1  0
...
383  333   1  1  0
  0    0   1  1  0
  2  550   2  0  0
…
993  766   2  0  0
491  235   2  0  1
  3  772   2  0  1
…
```

where the third column must be “1” for all points of the antecedent pattern 1 and “2” for all points of the qualitatively marked pattern 2, the forth column gives the value of the mark attached to the type 1 point (which is not used in our case, but got a value of 1), and the fifth column the value of the mark attached to the type 2 point. The fifth column yields “1” for the event (e.g., dead) and zero otherwise.

Trivariate random labeling based on mark correlation functions example Book_Fig3_27_mcf.res

1. Execute *Programita*.
2. Select “**Mark correlation functions**” in window What do you want to do?
3. Highlight data file (Book_Fig3_27.mcf) you want to analyze in Input data
4. Click “**List with coordinates, no grid**” in MCFunction
5. Provide in the window mark correlation functions the bin width in data units (1), an appropriate ring width (5), and a maximal distance *r* of the analysis (100).
6. Disable “**Normalize**” to get non-normalized mark correlation functions.
7. Click the small **“ok”** button the window mark correlation functions
8. Press button “**Calculate Index**” and *Programita* shows you the pattern and the summary functions:



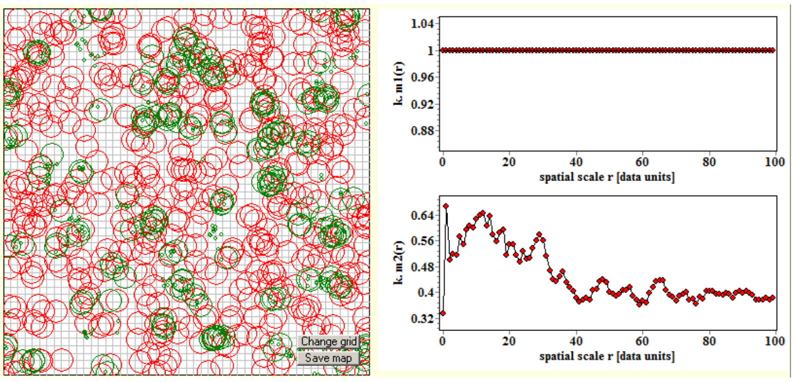
What do you want to do?

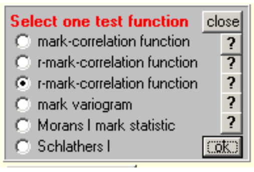
○ Standard analysis
● Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation

☑ Calculate simulation envelopes

Input data

Book_Fig2_16a.mcf
Book_Fig3_27.mcf

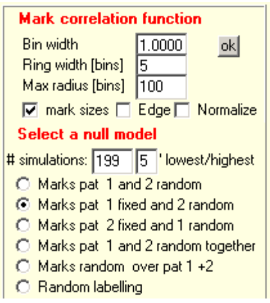




Select one test function

○ mark-correlation function ?
○ r-mark-correlation function ?
● r-mark-correlation function ?
○ mark variogram ?
○ Morans I mark statistic ?
○ Schlathers I

9. Select the third test function “r-mark correlation functions”
10. Check the checkbox “**Calculate simulation envelopes**”, select the number of simulations of the null model (199) and the rule for the simulation envelopes (5), and again “100” for the maximal radius
11. Select the null model “**Marks pat 1 fixed and 2 random**” which randomizes only the marks of the qualitatively marked pattern.
12. Press button “**Calculate Index**”.



Mark correlation function

Bin width 1.0000

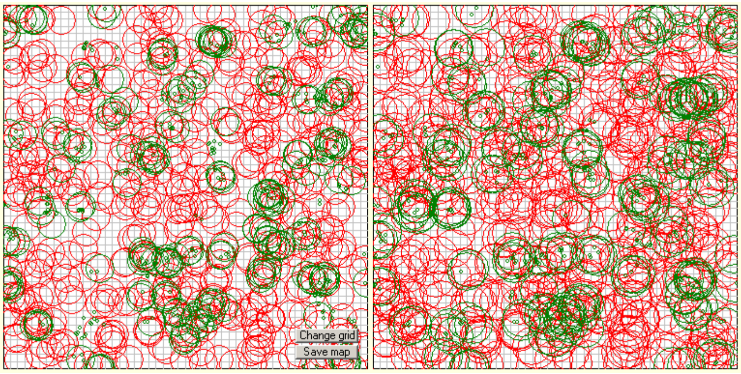
Ring width [bins] 5
Max radius [bins] 100
☑ mark sizes □ Edge □ Normalize

Select a null model

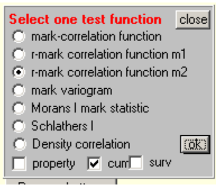
simulations: 199 5 ' lowest/highest

○ Marks pat 1 and 2 random
● Marks pat 1 fixed and 2 random
○ Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
○ Random labelling

13. *Programita* now shows the data and the simulations of the null model where the red circles are the points of the antecedent pattern 1 with radius proportionally to the mark (1) and the green circles are the points of the second pattern with a large circle for dead (mark 1) and a small circle for surviving (mark 0):

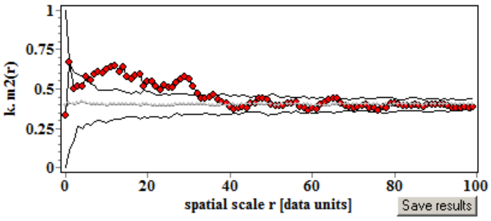


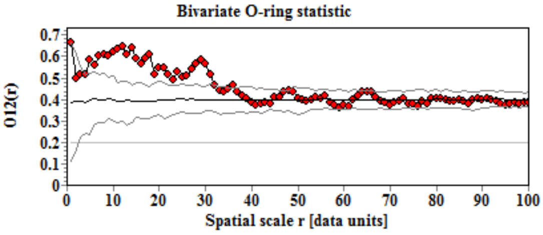
14. After termination of the simulations, *Programita* shows the results of the trivariate random labeling analysis (the r-mark correlation function m2) which are virtually identical with that of the grid-based analysis:


Select one test function
close
○ mark-correlation function
○ r-mark correlation function m1
● r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation
ok
□ property ☑ cum □ surv

mark correlation analysis:

grid-based analysis:





5.2.6 Random labeling for communities

The standard random labeling works for a univariate pattern that carries a qualitative mark. In some cases, however, the data are given by a multivariate pattern (e.g., the locations of all saplings in a tropical forest) and a qualitative mark such as surviving vs. dead. In this case we can run the standard random labeling analysis without regard to the species (i.e., the observed species-specific mortality rates are not conserved). This is called "**community-wide species-blind random labeling** in Wiegand and Moloney (2014)

However, another possibility is to conserve the observed species-specific mortality rates in the random labeling null model. Thus, conventional random labeling is conducted here within each species, but the final test statistics average over all species. This is called "**community-wide species-specific random labeling**" in Wiegand and Moloney (2014). Details can be found in section 4.4.2.3 Community Wide Random Labeling in Wiegand and Moloney (2014).

The coding of the data file is almost the same as for standard random labeling, but an additional fifth column given the species number is required:

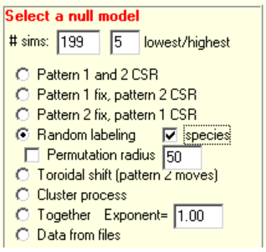
```
0  1000  0  500  2582
 37.76  308.91  0  1  3
 39.92  348.13  1  0  3
144.53   25.32  0  1  4
229.01   64.05  1  0  4
435.43  351.24  0  1  4
506.88  478.12  1  0  5
…
```

Community-wide species-specific random labeling, example Book_Fig4_42_species.res

To generate this example we first conducted pattern reconstruction with the underlying univariate pattern of the original data file of Figure 4.42 (the pattern of small saplings of gap species at the BCI forest) and then applied a mortality rate of 35% that depended linearly on the number of neighbors within 10m. Finally, the species labels of the original data where then randomly distributed over the points. Thus, the pattern shows density dependent mortality but there is no difference between species-specific and species-blind random labeling.

1. Select "**Standard analysis**" in window What do you want to do?
2. Highlight a data file in Input data ("Book_Fig4_42_rand.dat" in the example) and click the small "ok" button.
3. Select bin of 1m window Select a new cell size
4. Select a ring width of 5m in window "Which method will you use"
5. Accept selection of neighborhood ranks for estimation of $D^k(r)$ (will not be used in random labeling analyses).
6. Enable the checkbox "Calculate simulation envelopes" in the window What do you want to do? and select the random labeling null model in the Select a null model window.
7. Specify the **number of simulations of the null model** (199 in the example) and the rule for the estimation of **simulation envelopes** (here the 5th lowest and highest values of the summary function of the 199 null model data sets).

8. Check the checkbox “**species**” beside of “Random labeling”:

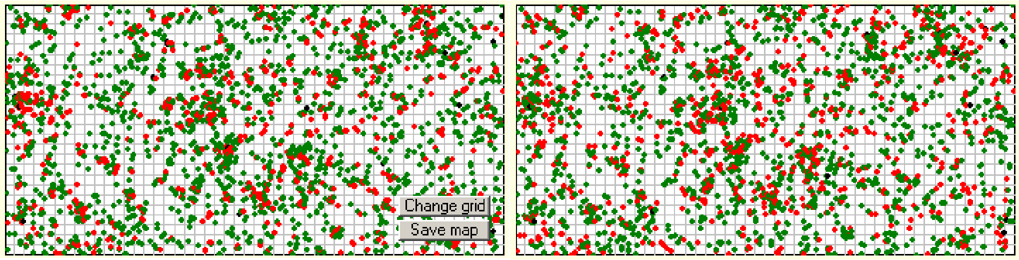


Select a null model

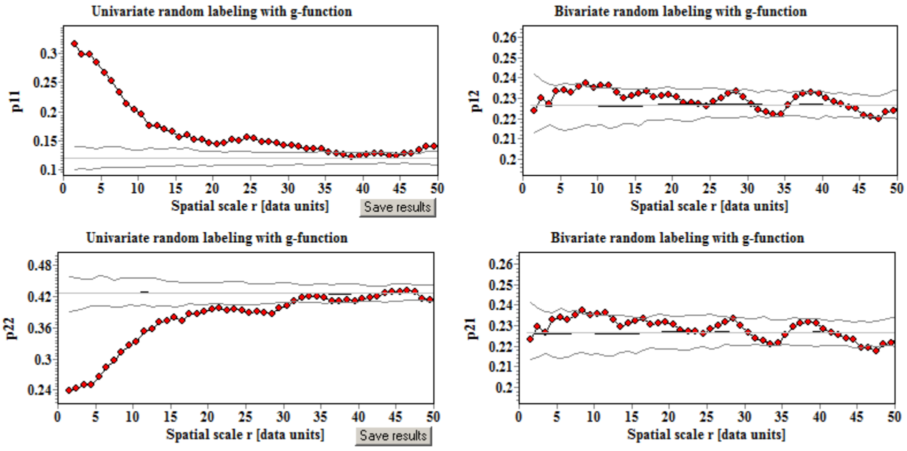
sims: 199 5 lowest/highest

○ Pattern 1 and 2 CSR
○ Pattern 1 fix, pattern 2 CSR
○ Pattern 2 fix, pattern 1 CSR
● Random labeling ☑ species
□ Permutation radius 50
○ Toroidal shift (pattern 2 moves)
○ Cluster process
○ Together Exponent= 1.00
○ Data from files

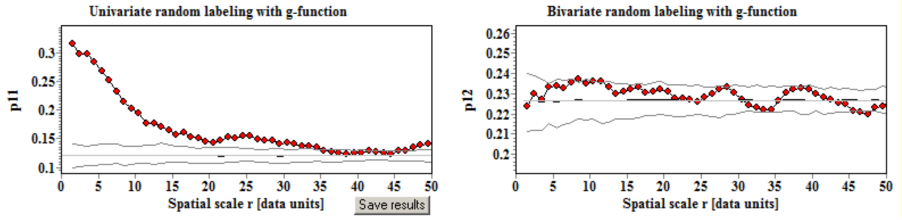
9. Press button “**Calculate Index**” and *Programita* shows the observed and simulated pattern:



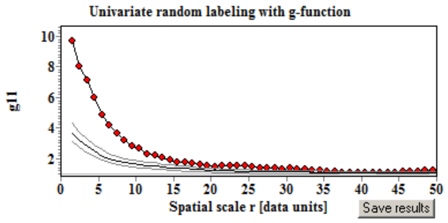
10. The results of the simulations of the species-specific random labeling null model show indeed that effects are not species specific. This is recognized because the expected mark connection functions of the null model (the black line) are horizontal lines with values of $p_1\ p_1$ in the univariate case and $p_1\ p_2$ in the bivariate case, with p_1 and p_2 being the proportion of type 1 and type 2 points among all points:

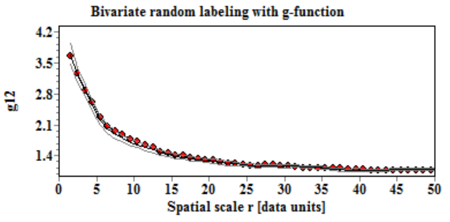


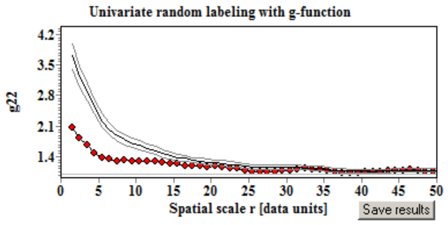
These expectations are identical to the expectation of community-wide species-blind random labeling (i.e., standard random labeling without regard to the species label; the grey line) (example Book_Fig4_42_rand.res):

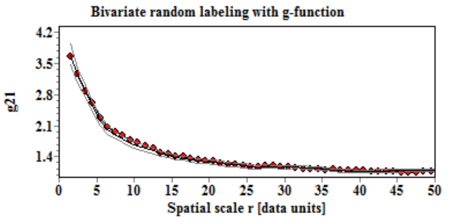


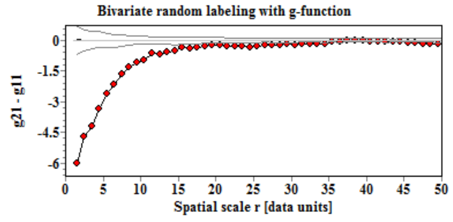
Community-wide species-specific random labeling. Example *Book_Fig4_42_species.res*

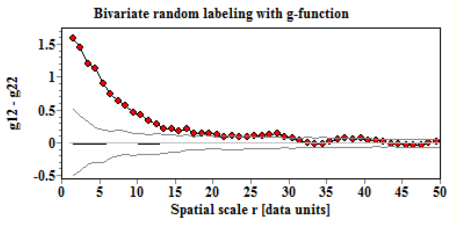


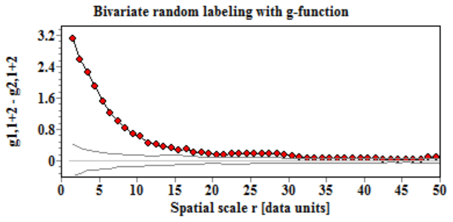


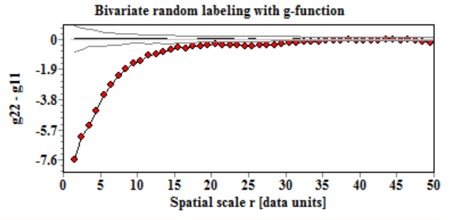










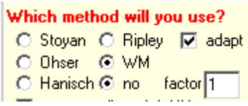


5.3 Combine replicates for random labeling

In some cases you may have maps of several replicate plots of a larger point pattern under identical conditions. In this case the resulting test statistics of the individual replicate plots can be combined into average test statistics (Diggle 2003: page 123; Illian et al. 2008: page 263; Wiegand and Moloney 2014: section 3.2). This is of particular interest if the number of points in each replicate plot is relatively low. In this case the simulation envelopes of individual analyses would become wide, but combining the data of several replicate plots into average test statistics increases the sample size and thus narrows the simulation envelopes. Section 3.2.1 of Wiegand and Moloney (2014) provides details on the aggregation formulas for different summary functions and section 3.2.2 several examples.

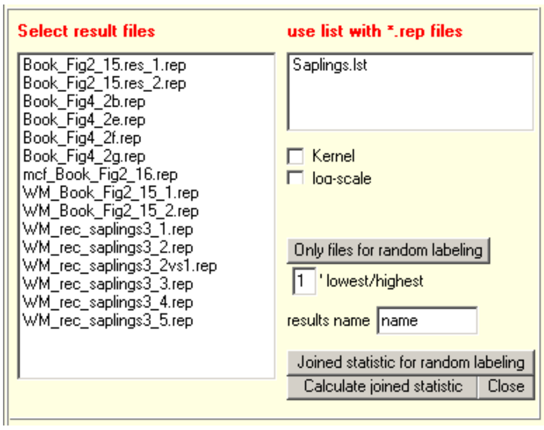
It is also possible to combine the random labeling analyses of all several species of a community into one function to get the community average.

The default estimators of the *Programita* standard mode (and the grid-based mode) use the WM estimators for the pair correlation and the *K*-function based on the quantities $\lambda g(r)$ and $\lambda K(r)$. The corresponding aggregation formulas for the WM estimator are provided in equations 3.114 and 3.117 in Wiegand and Moloney (2014). The specific test functions of random labeling analyses are also based on these estimators.


Which method will you use?
○ Stoyan ○ Ripley ☑ adapt
○ Ohser ● WM
○ Hanisch ● no factor 1

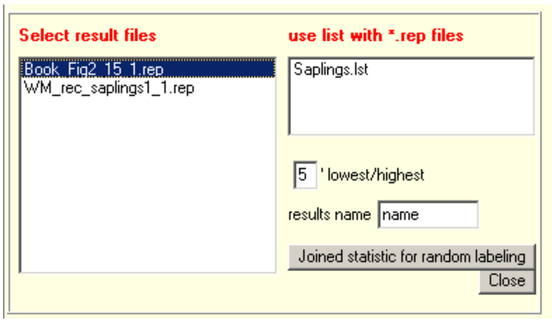
If the null model used in your analyses was random labeling, *Programita* saved two results files per analysis, a *_1.rep and a *_2.rep.

To simplify selection of results files and to tell *Programita* that you will combine replicates that used random labeling click “**Only files for random labeling**”. In this case *Programita* hides all *.rep files which are not random labeling.


Select result files
Book_Fig2_15.res_1.rep
Book_Fig2_15.res_2.rep
Book_Fig4_2b.rep
Book_Fig4_2e.rep
Book_Fig4_2f.rep
Book_Fig4_2g.rep
mcf_Book_Fig2_16.rep
WM_Book_Fig2_15_1.rep
WM_Book_Fig2_15_2.rep
WM_rec_saplings3_1.rep
WM_rec_saplings3_2.rep
WM_rec_saplings3_2vs1.rep
WM_rec_saplings3_3.rep
WM_rec_saplings3_4.rep
WM_rec_saplings3_5.rep

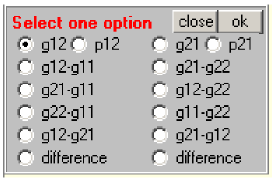
use list with *.rep files
Saplings.lst
□ Kernel
□ log-scale
Only files for random labeling
1 ' lowest/highest
results name name
Joined statistic for random labeling
Calculate joined statistic | Close

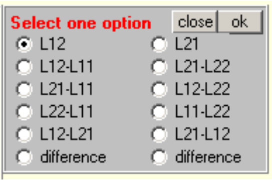
Now select the replicates as before, but click “**Joined statistic for random labeling**”.


Select result files
Book_Fig2_15_1.rep
WM_rec_saplings1_1.rep

use list with *.rep files
Saplings.lst
5 ' lowest/highest
results name name
Joined statistic for random labeling
Close

You can view the results of all random labeling test statistics by using the test statistics in the windows below.


Select one option close ok
● g12 ○ p12 ○ g21 ○ p21
○ g12-g11 ○ g21-g22
○ g21-g11 ○ g12-g22
○ g22-g11 ○ g11-g22
○ g12-g21 ○ g21-g12
○ difference ○ difference

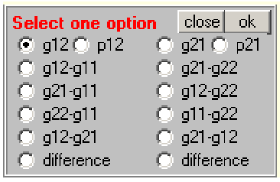

Select one option close ok
● L12 ○ L21
○ L12-L11 ○ L21-L22
○ L21-L11 ○ L12-L22
○ L22-L11 ○ L11-L22
○ L12-L21 ○ L21-L12
○ difference ○ difference

However, to obtain test functions based on the pair correlation function or the *K*-function you need to conduct separate analyses.

The “WM_name_1.rep” and “WM_name_2.rep” (or “R_name_1.rep” and “R_name_2.rep”) files for random labeling are basically the same as for the standard analysis mode:

```
   50     199   244     84 250000  W-M     1  gridless  g(r)
 0  0   3059.5516     2.0000      2.0000  164.10  472.77
 0  1   6864.6946     6.0000      3.0000  219.41  316.06
 0  2  11409.6905    10.0000      5.0000  220.01  316.94
 0  3  15918.2302    16.0000      4.0000  252.32  181.74
 0  4  20401.9507    18.0000      6.0000  221.47  212.69
 0  5  24862.8655    28.0000      7.0000  282.70  203.62
 0  6  29303.3423    26.0000     11.0000  222.73  271.49
 0  7  33725.9864    30.0000     10.0000  223.30  214.44
C1 C2  C3             C4           C5      C6      C7
```

The “WM_name_1.rep” contains the information on the $\lambda_1 g_{11}(r)$ and $\lambda_2 g_{12}(r)$ [or the $\lambda_1 K_{11}(r)$ and $\lambda_2 K_{12}(r)$] and the “WM_name_2.rep” contains the information on the $\lambda_2 g_{22}(r)$ and $\lambda_1 g_{21}(r)$ [or the $\lambda_2 K_{22}(r)$ and $\lambda_1 K_{21}(r)$] required to assemble the different test statistics to be selected.



Select one option close ok

● g12 ○ p12 ○ g21 ○ p21

○ g12-g11 ○ g21-g22

○ g21-g11 ○ g12-g22

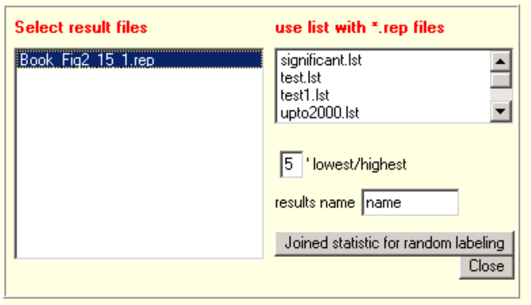
○ g22-g11 ○ g11-g22

○ g12-g21 ○ g21-g12

○ difference ○ difference

Change ring width for pair correlation function

If you use the pair correlation function as summary function, this feature of *Programita* allows you also to change a posteriori the ring width. **To take advantage of this feature, the original analysis must be done with ring width of 1**: 1 ring widht. For ring width = 1 the plot of the pair correlation function will be rugged.



Select result files

Book_Fig2_15_1.rep

use list with *.rep files

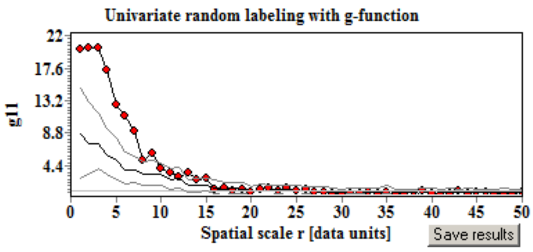
significant.lst
test.lst
test1.lst
upto2000.lst

5 ' lowest/highest

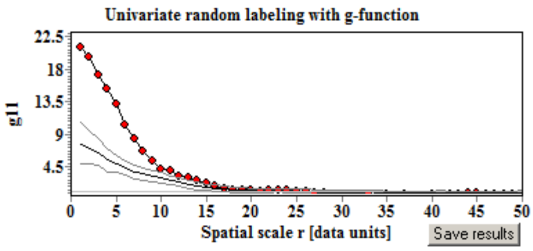
results name name

To select a posteriori a wider ring width select the file “Book_Fig2_15_1.rep” to read the results of the analysis of Figure 2.15 in Wiegand and Moloney (2014), and then go” again to “Replicates”. Now, when highlighting the again the file “Book_Fig2_15_1.rep” you can select a new ring width, for example a ring width dr = 5: Ring width [bin] 5 . See below the example with dr = 1 (left) and dr = 5 (right):

dr = 1



dr = 5



6 Analysis with mark correlation functions

Because the data structure of quantitatively marked patterns is different from that of the standard univariate and bivariate patterns, I introduced an own *.mcf extension for the data files of mark correlation analysis. *Programita* allows for mark correlation analysis of the following data structures:

- univariate patterns with one quantitative mark (data type 6)
- univariate patterns with two quantitative marks (data type 7)
- qualitatively marked patterns with one quantitative mark (data type 8)
- bivariate patterns with one quantitative mark (data type 9)

Data type 9 includes also trivariate random labeling as special case.

The estimators for the mark correlation functions are explained in detail in Section 3.1.7 “Summary functions for Quantitatively Marked Point Patterns” of Wiegand and Moloney (2014). All estimators for mark correlation functions included in *Programita* are based on real distances between pairs of points.

6.1 Test functions for mark correlation functions

The basic idea of mark correlation functions is simple. I illustrate it for the simplest data type 6 where each point i with coordinates $\mathbf{x}_i$ carries a mark of value m_i.

1. You identify all pairs i–j of points which are located approximately at distance r [this is defined by the box kernel; the pair belongs to distance class r if its distance is within the interval (r - dr/2, r + dr/2) where dr is the ring width]. Point i is the focal point, and point j the second point.
2. You estimate the mean value of a test function $t(m_i, m_j)$ of the marks m_i and m_j of the i–j point pair, respectively, taken over all i–j pairs determined in step 1.
3. You repeat step 2 for a range of distances r to obtain a non-normalized mark correlation function $c_t(r)$ where the “t” refers to the test function used.
4. Finally, the normalized mark correlation function is estimated as $k_t(r) = c_t(r)/c_t$ where the c_t is the average of the test function, but taken over ALL possible pairs of points i–j, regardless of their distance.

A wide variety of test functions can be constructed, depending on the specific objectives. Here are the ones implemented in *Programita*:

t	Name	Symbol	Test function	c_t
t_1	mark correlation function	$k_{mm}(r)$	$m_i\, m_j$	μ^2
t_2	r-mark correlation function	$k_{m\,.}(r)$	m_i	μ
t_3	r-mark correlation function	$k_{.\,m}(r)$	m_j	μ
t_4	mark variogram	$\gamma(r)$	$(m_i - m_j)^2/2$	σ^2
t_5	Moran's *I*	$I_{mm}(r)$	$(m_i - \mu)\ (m_j - \mu)$	σ^2
t_6	Schlather's *I*	$I_{mm}(r)$	$[m_i - \mu(r)][m_j - \mu(r)]$	σ^2
t_7	Cumulative density correlation function	$C_{mK}(r)$†*	$(m_i - \mu)[(\lambda K_i(r) - \lambda K(r)]/(\pi r^2)$	$\sigma\, \sigma_K$
t_8	Non-cum. density correlation function	$C_{mg}(r)$†*	$(m_i - \mu)[(\lambda g_i(r) - \lambda g(r)]$	$\sigma\, \sigma_g$
t_9, t_{10}	Strength of density on mortality	$dd_1(r)$‡*	-	μ

†these functions are always normalized

‡ applies only for the special case if the quantitative mark m_i is binary with values 1 and 0.

* not presented in Wiegand and Moloney (2014)

where

m_i	mark of focal point i
m_j	mark of second point j
μ	mean of the marks taken over all points
$\mu(r)$	$c_2(r)$: the non-normalized r-mark correlation function; the conditional mean of the mark of the first point, taken over all pairs of points located at distance r
λ	the intensity of the pattern
$\lambda K_i(r)$	the "local" K-function; the number of points within distance r of point i
$\lambda g_i(r)$	the "local" g-function; the density of points within distance r of point i
σ^2	the variance of the marks m_i
σ_K	the variance of $\lambda K_i(r)$
σ_g	the variance of $\lambda g_i(r)$

Mark correlation function

- The mark correlation function $k_{mm}(r)$ is the normalized mean of the mark product $t(m_i, m_j) = m_i\, m_j$ of all i–j pairs of points distance r apart.

If the marks are randomly distributed over the points, the mean mark product does not differ from the mean mark product taken over all pairs of points (i.e., $c_1 = \mu^2$) and we find $k_{mm}(r) = 1$. However, if the marks of nearby points are consistently smaller than the mean mark we find $k_{mm}(r) < 1$ (inhibition), and conversely, if the marks of nearby points are consistently larger than the mean mark we find $k_{mm}(r) > 1$ (mutual stimulation).

r-mark correlation function

To estimate the r-mark correlation function we first determine all pairs of points of the pattern that have an approximate distance r. The marks of the kth pair are (m_{1k}, m_{2k}) where m_{k1} is the mark of the first point and m_{k2} the mark of the second point. The number of pairs at distance r is $n(r)$.

- The r-mark correlation function $k_{m\,.}(r)$ corresponds to test function $t_2(m_{1k}, m_{2k}) = m_{1k}$ and is the normalized mean of m_{1k}, taken over all $n(r)$ pairs of points distance r apart.
- The r-mark correlation function $k_{.\,m}(r)$ corresponds to test function $t_3(m_{1k}, m_{2k}) = m_{2k}$ and is the normalized mean of m_{2k}, taken over all $n(r)$ pairs of points distance r apart.

The non-normalized r-mark correlation functions for data type 6 are also termed $\mu(r)$ and the normalization constants are given by $c_2 = c_3 = \mu$; they are the mean μ of the marks, taken over all points of the pattern. Later in the bivariate cases we will find subtle differences between the two r-mark correlation functions.

Schlather's *I*

Schather's I is the standard Pearson cross-correlation coefficient of the data set (m_{1k}, m_{2k}). Thus, we first estimate the mean of the two entries (m_{1k}, m_{2k}), that both yield $\mu(r)$ for data type 6, and then estimate the covariance $[m_{1k} - \mu(r)][m_{2k} - \mu(r)]$ which is then normalized with the product of the standard deviations (which both yield σ) to yield the standard cross-correlation coefficient.

Moran's *I*

Note that older studies use instead of the Schlather test function t_6 the test function t_5 with $t_5(m_{1k}, m_{2k}) = (m_{1k} - \mu)\ (m_{2k} - \mu)$ that is based on the global mark mean μ, and not on the conditional mark mean $\mu(r)$ estimated from the pairs of points of the data set (m_{1k}, m_{2k}). Departures from this function may be strongly impacted by departures from the r-mark correlation function and not due to correlations in the marks of the pairs of points (m_{1k}, m_{2k}).

The mark variogram

- The mark variogram $\gamma(r)$ is based on the test function $t_4(m_{1k}, m_{2k}) = (m_{1k} - m_{2k})^2/2$, half of the squared difference between the marks of the kth point pair.

It can be used to assess if nearby points have more similar or dissimilar values in their marks. If the values of m_{1k} and m_{2k} are more similar than expected by the overall variance σ^2 in the marks, the mark variogram $\gamma(r)$ will have values smaller than one and if the values of the two marks are more dissimilar than expected, we find $\gamma(r) > 1$.

Density correlation functions

I developed the (cumulative) density correlation functions for the study of Fedriani et al. (2015) to assess the correlation between the reproductive success of trees and the density of neighbors located within distance r. The density correlation functions are the standard Pearson cross-correlation coefficient of the data sets $[m_i, \lambda K_i(r)/(\pi r^2)]$ or $[m_i, \lambda g_i(r)]$, where m_{1i} is the mark of point i and the constructed marks $\lambda K_i(r)/(\pi r^2)$ and $\lambda g_i(r)$ of point i give the density of density of neighbors of point i within and at distance r, respectively.

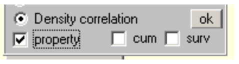
Because the correlation coefficients are independent on multiplication of the data with a constant, we can simplify the test functions to

$$t_7(m_i, K_i(r)) = (m_i - \mu)[(K_i(r) - K(r)]$$
$$t_8(m_i,\ g_i(r)) = (m_i - \mu)[(g_i(r) - g(r)]$$

The normalization constants are then the products $\sigma\ \sigma_K$ and $\sigma\ \sigma_g$ of the standard deviations σ and σ_g (and σ and σ_K) of the data sets $[m_{1i}, K_i(r)]$ and $[m_{1i}, g_i(r)]$, respectively.

Density correlation functions with sum of marks instead of number of neighbors

In some cases not only the number of neighbors, but also their marks may have effects on the value of the mark of the focal point. The option “property” considers this for the density correlation functions: it uses the sum of the marks of the neighbors instead of the number of neighbors for the correlation analysis. In the univariate case the sum of the marks of the neighbors of pattern 1 are used, and in the bivariate case the sum of the second mark (data type 7) or the sum of the quantitative marks of the second pattern (data types 8 and 9).



● Density correlation

☑ property □ cum □ surv

Strength of density on mortality

Some of the mark correlation functions can also be applied for a binary mark such as surviving (s, mark 1) vs. dead (d, mark 0). For example, in this case the non-normalized *r*-mark correlation function $c_2(r)$ yields the mean proportion of focal points that survive (*s*), taken over all pairs of points separated by distance *r*. *Programita* uses a transformation $dd_1(r)$ of this function (with direct relationship to the densities of neighboring points), instead of the density correlation functions if the mark m_i is a binary mark with values 1 or 0. The univariate and non-cumulative function $dd_1(r)$ is the ratio

$$dd_1(r) = g_{\mathrm{d,s+d}}(r)/g_{\mathrm{s,s+d}}(r)$$

where $g_{\mathrm{d,s+d}}(r)$ is the partial pair correlation function giving the total density of points (i.e., with marks surviving and dead; s+d) around points with the mark dead (d) and the $g_{\mathrm{s,s+d}}(r)$ is the partial pair correlation function giving the corresponding density of points (s+d) around points with the mark surviving (s).

Analogously, the cumulative function $DD_1(r)$ is the ratio

$$DD_1(r) = K_{\mathrm{d,s+d}}(r)/K_{\mathrm{s,s+d}}(r)$$

where $K_{\mathrm{d,s+d}}(r)$ and $K_{\mathrm{s,s+d}}(r)$ are the corresponding partial *K*-functions.

As we will show below, the non-normalized *r*-mark correlation function $c_2(r)$ is closely related with $dd_1(r)$:

$$c_2(t) = s\frac{1}{dd(r)(1-s)+s}$$

where *s* is the proportion of points with mark surviving (= the mean mark μ if the mark for surviving is one and that for dead zero). Thus, departures of the non-normalized *r*-mark correlation function $c_2(r)$ from the expected survival rate *s* occur only because of density dependence in mortality.

Note that the bivariate $dd_2(r)$ of data type 9 is especially interesting because it allows to quantify the effect of a second pattern on the quantitative mark of focal pattern 1.

Extension of mark correlation functions to other data types

“Bivariate” mark correlation functions can be derived for different data types where the points carry either two marks or two types of points carry one mark each. *Programita* allows for mark correlation analysis of the following “bivariate” data structures:

- one type of points with two quantitative marks (data type 7)
- two types of points (a qualitative mark) with one quantitative mark (data type 8)
- two types of points (a bivariate pattern) with one quantitative mark (data type 9)

Data type 7 (section 3.1.7.3)

In this case each point i carries two marks m_{i1}, m_{i2}. The bivariate test functions are then based on the first mark of point i and the second mark of point j (m_{i1}, m_{j2}). In this case we need the mean marks μ_1 and μ_2, the conditional means $\mu_1(r)$ and $\mu_2(r)$, and the covariance σ^2_{12} of the two marks. Note that in case of the density correlation functions and $dd_2(r)$, the "univariate" and "bivariate" functions are the same if the option "property" is not used. Otherwise they estimate the correlation between the first mark m_{i1} of point i and the sum of the first marks m_{j1} of points within (or at) distance r (univariate) and the sum of the second marks m_{j2} of points within (or at) distance r (bivariate).

t	Name	Symbol	bivariate test functions	c_t
t_1	mark correlation function	$k_{m1m2}(r)$	$m_{i1}\ m_{j2}$	$\mu_1\mu_2$
t_2	r-mark correlation function	$k_{m1\ .}(r)$	m_{i1}	μ_1
t_3	r-mark correlation function	$k_{.\ m2}(r)$	m_{j2}	μ_2
t_4	mark variogram	$\gamma_{m1m2}(r)$	$(m_{i1}-m_{j2})^2/2$	σ_{12}^2
t_5	Moran's I	$I_{m1m2}(r)$	$(m_{i1}-\mu_1)\ (m_{j2}-\mu_2)$	σ_{12}^2
t_6	Schlather's I	$I_{m1m2}(r)$	$[m_{i1}-\mu_1(r)][m_{j2}-\mu_2(r)]$	σ_{12}^2
t_7	cum. density correlation function	$C_{m1K}(r)$	$(m_{i1}-\mu_1)[(K_i(r)-K(r)]$*	$\sigma\ \sigma_K$
t_8	non-cum. density correlation function	$C_{m1g}(r)$	$(m_{i1}-\mu_1)[(g_i(r)-g(r)]$*	$\sigma\ \sigma_g$
t_9, t_{10}	strength of density on mortality	$dd_2(r)$	-	μ_2

* the bivariate test functions use the marks of m_{i2} of the neighboring points if the option "property" is enabled.

Data type 8 (section 3.1.7.4)

In this case each point i carries one qualitative mark (e.g., type 1 for surviving vs. type 2 for dead) and one quantitative mark (e.g., size). In this data structure the focal points are of type 1 and the second point of the pair is of type 2. Thus, the quantitative marks of point pair i–j are m_{1i} and m_{2j}. Two null models are possible here, randomization of the qualitative mark over the points or randomization of the quantitative mark over the points. The corresponding estimators of the mark correlation functions differ slightly.

1) Randomize quantitative mark

In this case the null hypothesis is that type 1 and type 2 points do not differ in their quantitative marks. Thus, the marks m_{1i} and m_{2j} are randomly shuffled over the joined pattern of type 1 and type 2 points. We therefore need to normalize with μ_{1+2}, the mean mark of the joined pattern of type 1 and type 2 points. Similarly, we use the conditional mean $\mu_{1+2}(r)$ and the variance σ^2_{1+2} of the joined pattern of type 1 and type 2 points in the test functions and normalization. The univariate mark correlation functions use pairs of type 1 – type 1 points whereas the bivariate mark correlation functions use pairs of type 1 – type 2 points.

t	Name	Symbol	bivariate test functions	c_t
t_1	mark correlation function	$k_{m1m2}(r)$	$m_{1i}\ m_{2j}$	μ^2_{1+2}
t_2, t_3	r-mark correlation function	$k_{m1\ .}(r)$	m_{1i}, m_{j2}	μ_{1+2}
t_4	mark variogram	$\gamma_{m1m2}(r)$	$(m_{1i}-m_{2j})^2/2$	σ_{1+2}^2
t_6	Schlather's I	$I_{m1m2}(r)$	$[m_{1i}-\mu_{1+2}(r)][m_{j2}-\mu_{1+2}(r)]$	σ_{1+2}^2
t_7	cum. density correlation function	$C_{m1K2}(r)$	$(m_{1i}-\mu_{1+2})[(K_{2i}(r)-K_2(r)]$	$\sigma\ \sigma_K$
t_9, t_{10}	strength of density on mortality	$dd_2(r)$		-

2) Randomize qualitative mark (random labeling)
In this case the membership to type 1 and type 2 is changed in every simulation of the null model. Therefore we estimate the mean mark μ_1 over type 1 points and the mean mark μ_2 over type 2 points, as well as the conditional means $\mu_1(r)$ and $\mu_2(r)$, and the covariance σ_{12} of the marks among all type 1 and type 2 points. In contrast to the other cases, the values of μ_1, μ_2 and σ_{12} must be estimated after each simulation of the null model. As above, the univariate mark correlation functions use pairs of type 1 – type 1 points whereas the bivariate mark correlation functions use pairs of type 1 – type 2 points.

t	Name	Symbol	bivariate test functions	c_t
t_1	mark correlation function	$k_{m1m2}(r)$	$m_{1i}\ m_{2j}$	$\mu_1\mu_2$
$t_{2,}\ t_3$	r-mark correlation function	$k_{m1}\ .(r)$, $k_{m2}\ .(r)$	m_{1i}, m_{j2}	μ_1, μ_2
t_4	mark variogram	$\gamma_{m1m2}(r)$	$(m_{1i}–m_{2j})^2/2$	σ_{12}
t_6	Schlather's I	$I_{m1m2}(r)$	$[m_{1i}–\mu_1(r)][m_{j2}–\mu_2(r)]$	σ_{12}
t_7	cum. density correlation function	$C_{m1K}(r)$	$(m_{1i}–\mu_1)[(K_{2i}(r)–K_2(r)]$	$\sigma\ \sigma_K$
t_9, t_{10}	strength of density on mortality	$dd_2(r)$	-	-

Data type 9 (section 3.1.7.5)
In this case we have two patterns (type 1 and type 2) where each point i of pattern1 and each point j of pattern 2 carries one quantitative mark. Thus, the data structure comprises two types of points with one mark each. The focal points are of type 1 and the second points of type 2. Thus the marks of point pair i–j are m_{1i} and m_{2j}. In this case the quantitative marks m_{1i} and m_{2j} are randomly shuffled within patterns 1 and/or 2. We therefore normalize with the mean mark μ_1 of type 1 points and the mean mark μ_2 of type 2 points and the covariance σ_{12} between the marks of the two patterns. The univariate mark correlation functions use pairs of type 1 - type 1 points whereas the bivariate mark correlation functions use pairs of type 1 - type 2 points. The estimators are as above for data type 8 and the random labeling null model.

6.2 Analysis of univariate patterns with one mark (data type 6)

Univariate quantitatively marked patterns comprise the coordinates of a univariate pattern, but each point carries an additional quantitative mark that characterizes the ecological object that is idealized as point. The quantitative mark is usually a continuous attribute such as the size of a tree, but can also be an integer such as the number of seeds of a tree.

The major interest in analysis of univariate quantitatively marked patterns is in revealing potential spatial correlation structure of the marks, conditional on the underlying univariate pattern. For example, are trees which are closer together usually smaller than the average tree, or is fruit initiation higher for trees with more neighbors at small distances (Fedriani et al. 2015)? The basic null model for this data type is the so-called **independent marking null model** which randomly shuffles the marks over the points, thus removing all potential spatial structure in the marks. Section 3.1.7.1 in Wiegand and Moloney (2014) provides examples for the different analyses of qualitatively marked patterns that are useful in ecology. Fedriani et al. (2015) uses mark correlation functions, and especially the density correlation function, to show how density dependent effects in the different stages of plant reproductive success operate either in the same or in different directions and thus reinforce or neutralize each other.

6.2.1 Data preparation for data type 6

The data files must be an ASCII file with the *.mcf extension with the following format (the example shows the first lines of the data file Book_Fig2_16a.mcf):

```
0   500   0   500   600
249.75  451.80  1  3.725  0
434.30  272.60  1  4.726  0
482.65   35.20  1  1.826  0
100.10  196.60  1  1.983  0
245.00  117.45  1  2.012  0
423.40   38.05  1  2.552  0
222.85  349.05  1  1.651  0
 32.15   21.55  1  3.040  0
208.65  352.10  1  4.177  0
 92.95  214.85  1  2.094  0
112.40  452.65  1  3.256  0
…
```

where first line gives the size of the observation window (500 × 500 units in the example) and the number of points in the pattern.

- the first two columns of the following lines are the coordinates of the points, the third column indicates the pattern and must be "1" for all points because this data structure is based on an univariate pattern.
- The forth column carries the mark of the point.
- The fifth column is reserved for a second mark or for the mark of a second pattern and must be therefore "0".

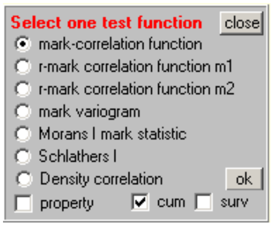
The data file must be a space or tab delimited ASCII file with the *.mcf extension. If you use Excel, there is a simple, but obviously generally unknown, way of saving files of a given type with a given extension:

1. Prepare the data file in Excel following the instructions above.
2. Then save as a tab delimited text file, but write "name.mcf" for the name (usually you would only write name and end up with a file named name.txt). The quotation marks are important because they force Excel to save the comma delimited file under the name name.mcf.

6.2.2 Steps data type 6 (Book_Fig2_16.res)

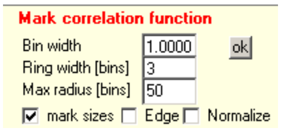
Programita estimates for data files of the *.mcf type several adapted test statistics based on mark correlation functions which are described in detail in section 3.1.7.1 “Univariate Quantitatively Marked Pattern (Data Type 6)” of Wiegand and Moloney (2014):

- the mark correlation function $k_{\text{mm}}(r)$
- the r-mark correlation function $k_{\text{m}\,.}\ (r)$
- the r-mark correlation function $k_{\,.\,\text{m}}(r)$
- the mark variogram $\gamma_{\text{mm}}(r)$
- a Moran’s I type mark statistics $I_{\text{mm}}(r)$
- Schlather’s correlation function $I_{\text{mm}}(r)$
- the density correlation functions $C_{m,g}(r)$ or $C_{m,K}(r)$



Select one test function
close
● mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation
ok
☐ property ☑ cum ☐ surv

The default for mark correlation functions is to use the normalized functions, i.e., $k_{\text{t}}(r) = c_{\text{t}}(r)/c_{\text{t}}$ where c_{t} is the normalization constant for a given test function t. However, you can select the non-normalized mark correlation functions $c_{\text{t}}(r)$ by disabling the checkbox “**Normalize**”. Note that the density correlation functions are always normalized.

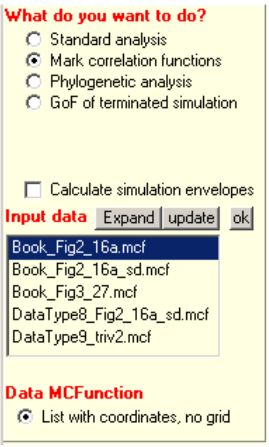


Mark correlation function
Bin width 1.0000
ok
Ring width [bins] 3
Max radius [bins] 50
☑ mark sizes ☐ Edge ☐ Normalize

Note that the two correlation coefficients $I_{\text{mm}}(r)$ may show absolute values larger than one if the number of point pairs at this distance is very low (say < 10). To avoid this increase the ring width.

The mark correlation mode can be accessed with the following sequence of actions:

1. Select “**Mark correlation functions**” in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file “Book_Fig2_16a.mcf”. The pattern is a superposition of 100 random points with a Thomas process of 100 clusters with an approximate radius 2σ of 10m. The mark attached to a point is proportional to one over the number of neighbors within 10 m.
3. Click “**List with coordinates, no grid**” in MCFunction
4. Provide in the window mark correlation functions the bin width in data units, an appropriate ring width, and a maximal distance r of the analysis. Select in the example a bin of 1m, a ring width of 3m, and a maximal distance of 50m.
5. If you use a ring width of 1 unit, you can later use the function “**Combine replicates**” to load the results of the analysis, to change the ring width, and to use the corresponding cumulative summary function (see below “View results of mark correlation analysis”).

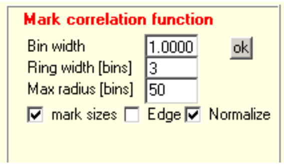


What do you want to do?
○ Standard analysis
● Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☐ Calculate simulation envelopes
Input data
Expand
update
ok

Book_Fig2_16a.mcf
Book_Fig2_16a_sd.mcf
Book_Fig3_27.mcf
DataType8_Fig2_16a_sd.mcf
DataType9_triv2.mcf

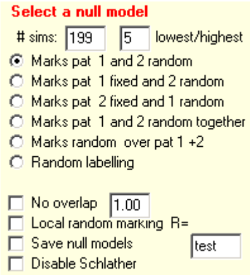
Data MCFunction
● List with coordinates, no grid

6. Disable “**Normalize**” if you want to use the non-normalized mark correlation functions. The default is “Normalize”
7. Check “**Edge**” if you want to use the Ripley edge correction. Default is no edge correction. Note that edge correction is not required for mark correlation functions.



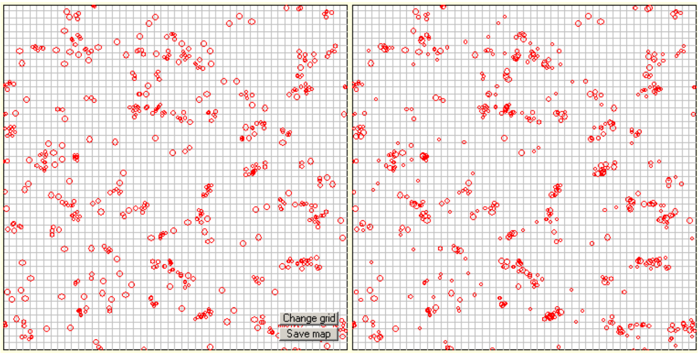
Mark correlation function
Bin width 1.0000
ok
Ring width [bins] 3
Max radius [bins] 50
☑ mark sizes ☐ Edge ☑ Normalize

8. Check "Calculate simulation envelopes" in window What do you want to do? The subwindow "Select a null model" appears.
9. Select an appropriate null model, the number of simulations of the null model (199), and the rule for the simulation envelopes (5' lowest and highest). Select for univariate patterns with one quantitative mark the null model "**Marks pat 1 and 2 random**" that shuffles the mark randomly over the points of the pattern. Note that not all null models are appropriate for all mark correlation function data type and that they must be selected with care.

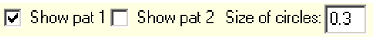


Select a null model
# sims: 199 5 lowest/highest
● Marks pat 1 and 2 random
○ Marks pat 1 fixed and 2 random
○ Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
○ Random labelling
□ No overlap 1.00
□ Local random marking R=
□ Save null models test
□ Disable Schlather

10. The estimation of Schlather's I and of the density correlation functions requires double calculations (because first the averages $\mu(r)$ and $g(r)$ need to be estimated); if you do not need this summary function you can speed up *Programita* by clicking "**Disable Schlather**".
11. Press button "**Calculate Index**" and *Programita* shows the observed and simulated pattern. The area of the disk that represents a point is proportionally to the mark:

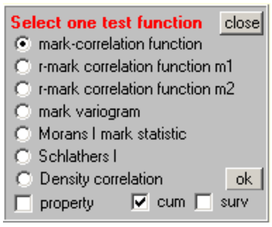


You notice that the points do not change their location in the null model simulation (right), but that the size of the points changes because the mark of all points is randomly shuffled. You can change the size of the circles by enlarging the factor in

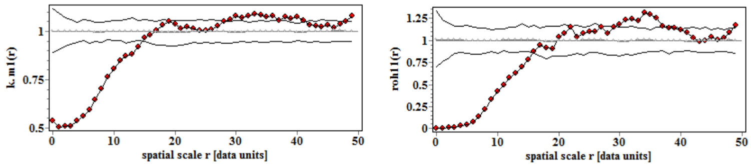


☑ Show pat 1 □ Show pat 2 Size of circles: 0.3

12. Use the radio buttons of the window Select one test function to select a mark correlation function and click the small "**ok**" button to get the result graphic:



Select one test function close
● mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation ok
□ property ☑ cum □ surv



The graph for the bivariate mark correlation is empty because the data were univariate. The results show that the mark of a point which is located at distance r from another point of the pattern is for distances $r < 16$m smaller than expected (r-mark correlation function; left) and that two points that are located closer than 16m have marks which are more similar than expected (mark variogram; right).

The density correlation functions

The non-cumulative density correlation function $C_{m,g}(r)$, is the standard Pearson correlation coefficient between the mark m_i of points i and the density of neighbors surrounding these points at distance r [$= \lambda g_\mathrm{i}(r)$]. Thus, the non-cumulative density correlation function is based on the test function:

$$t(r, m_\mathrm{i}, g_\mathrm{i}) = [m_\mathrm{i} - \mu][(\lambda g_\mathrm{i}(r) - \lambda g(r)]$$

where m_i is the mark of the focal point i, μ is the mean mark, λ the overall density of points in the observation window, $\lambda g_\mathrm{i}(r)$ the density of points at distance interval (r-dr/2, r+dr/2) of focal point i where dr is the width of the ring with radius r centered in point i, and $\lambda g(r)$ the mean density of points within these rings. The $g_\mathrm{i}(r)$ is the "local" pair correlation function and $g(r)$ the well known pair correlation function (Wiegand and Moloney 2014). Because correlation coefficient are independent on constants we can simplify the test function to

$$t(r, m_\mathrm{i}, g_\mathrm{i}) = [m_\mathrm{i} - \mu][(g_\mathrm{i}(r) - g(r)]$$

Programita uses here the Ohser edge correction weight (see Wiegand and Moloney 2014, section 3.1.2.1). The $C_{m,g}(r)$ is normalized by the product $\sigma_\mathrm{m}\sigma_\mathrm{g}$ of the standard deviations of the marks m_i and the local pair correlation functions $g_\mathrm{i}(r)$, respectively. The "*C*" in $C_{m,g}(r)$ stands for correlation, "*m*" for the first mark m_i, and "*g*" for the second mark $g_\mathrm{i}(r)$.

The analogous cumulative density correlation function $C_{m,K}(r)$ uses the *K*-function instead of the pair correlation functions:

$$t(r, m_\mathrm{i}, g_\mathrm{i}) = [m_\mathrm{i} - \mu][(K_\mathrm{i}(r) - K(r)]$$

where $K(r)$ is the *K*-function and $K_\mathrm{i}(r)$ the corresponding "local" *K* function. Additionally *Programita* estimates the correlation coefficient r_mnn between the mark m_i and the distance d_i to the nearest neighbor of point i based on the test function:

$$t(m_\mathrm{i}, d_\mathrm{i}) = [m_\mathrm{i} - \mu][[d_\mathrm{i} - d]$$

where d is the mean distance to the nearest neighbor. The result of the nearest neighbor correlation is written in the *.res results file at the last distance bin:

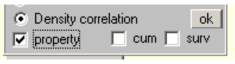
```
Scale r r    cm1d1(r)  E11-       E11+      mean11    cm1d2(r)  E12-       E12+      mean12
47.5 r r     0.085    -0.091      0.090     0.003     0.000     0.000      0.000     0.000
48.5 + r     0.116    -0.091      0.089     0.003     0.000     0.000      0.000     0.000
49.5 + r     0.788    -0.070      0.099     0.004     0.000     0.000      0.000     0.0000

results show density correlation function, but result for last distance bin r give correlation
between the nearest neighbor distance of a point and the value of the mark
```

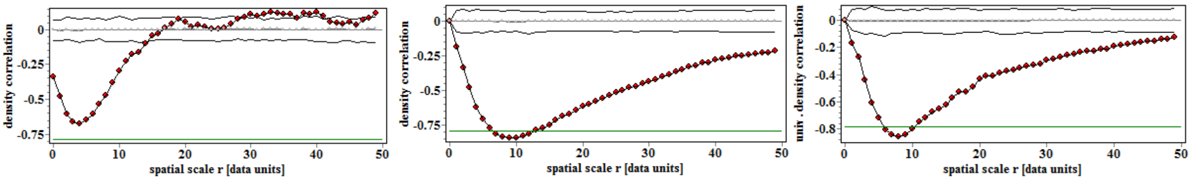
To switch between the cumulative and non-cumulate density correlation function you can use the small checkbox ☑ cum at the bottom of the window "Select one test function".

Density correlation functions with sum of marks instead of number of neighbors

In some cases not only the number of neighbors, but also their marks may have effects on the value of the mark of the focal point. The option "property" considers this for the density correlation function: it uses the sum of the marks of the neighbors instead of the number of neighbors for the correlation analysis.



The density correlation functions, for example for *Book_Fig2_16.res* are:



non-cumulative

cumulative

cumulative and “property”

Note that the mark attached to a point was in the example pattern proportional to one over the number of neighbors within a distance of 10 m. As expected, the cumulative density correlation function shows a very high negative correlation at distance 10m. Because the mark was only related to the number of neighbors, but not their sizes, the “property” option does not increase the correlation. The correlation coefficient r_{mnn} with the distance to the nearest neighbor is shown in the results graphs as a green line. Because the correlation is inverse (i.e., closer neighbors mean larger density) we show the negative value of r_{mnn}. In this example, the correlation of the mark with the distance to the nearest neighbor is almost as strong as that with the density of neighbors within distance r = 10m.

The density correlation functions for a binary mark (example *Book_Fig2_16a_sd.res*)

Some of the mark correlation functions can also be used for a binary mark (e.g., surviving vs. dead). For example, if dead is represented by mark 0 and surviving by mark 1, the non-normalized mark correlation function $c_1(r)$ yields the probability that of a pair of points separated by distance r both points are surviving. This is the well known mark connection function $p_{11}(r)$ (see example *Book_Fig_2_15.res*).

More interesting with respect to effects of density of neighboring points on mortality is the non-normalized r-mark correlation function $c_2(r)$, the proportion of focal points that survive taken over all pairs of points separated by distance r. As we will see, $c_2(r)$ is related to the density of points surrounding dead relative to surviving points. An estimator of $c_2(r)$ is

$$\hat{c}_2(r) = \frac{\sum_{i=1}^{n}\sum_{j=1,\neq}^{n} m_i k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)}{\sum_{i=1}^{n}\sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)}$$

where the mark m_{i} has value zero for dead and value one for surviving. We assume that the points are ordered in a way that the first n_{s} points have the mark surviving (i.e., m_{i} = 1) and the following $n - n_{\mathrm{s}}$ points have the mark dead (i.e., m_{i} = 0). Thus we find:

$$\hat{c}_2(r) = \sum_{i=1}^{n_s}\sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r) / [\sum_{i=1}^{n_s}\sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r) + \sum_{i=n_s+1}^{n}\sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)],$$

and by dividing with the enumerator, multiplying the sums with $n_{\mathrm{s}}/n_{\mathrm{s}}$ and $(n - n_{\mathrm{s}})/(n - n_{\mathrm{s}})$, and re-ordering we find that $c_2(r)$ contains the ratio of two partial pair correlation functions:

$$\hat{c}_2(r) = 1/[1 + \tfrac{n-n_s}{n_s}\frac{\frac{1}{n-n_s}\sum_{i=n_s+1}^{n}\sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)}{\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{j=1,\neq}^{n} k(\|\mathbf{x}_i - \mathbf{x}_j\| - r)}]$$

$$\hat{c}_2(r) = \frac{1}{1 + \frac{1-s}{s}\frac{g_{d,s+d}(r)}{g_{s,s+d}(r)}} = \frac{s}{s + (1-s)dd_1(r)}$$

All constants of the estimator of the pair correlation functions, except 1/ n_{s} and 1/($n - n_{\mathrm{s}}$), cancel. Therefore we have the ratio of the $g_{\mathrm{d,s+d}}(r)$, the partial pair correlation function giving the density of surviving and dead points (s+d) around dead points (d) and the $g_{\mathrm{s,s+d}}(r)$, the partial pair correlation function giving the density of surviving and dead points (s+d) around surviving points (d).

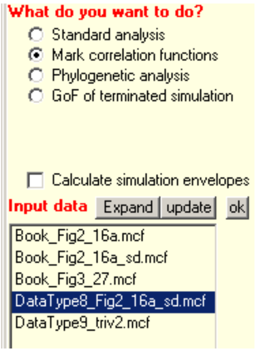
The function $dd_1(r)$ therefore gives the density of points (i.e., s+d) at distance r around dead points divided by the density of points at distance r around surviving points, i.e., $dd_1(r) = g_{\mathrm{d,s+d}}(r)/g_{\mathrm{s,s+d}}(r)$.

The function $dd_1(r)$ is therefore a measure of density effects on mortality: it indicates how many more neighbors dead points have on average relative to surviving points. It estimates the strength and direction of density effects on mortality. If $dd_1(r) > 1$ the effect of neighbors on survival is negative and for $dd_1(r) < 1$ the effect is positive. *Programita* uses the $dd_1(r)$ therefore instead of the density correlation function if the mark is binary mark (e.g., surviving vs. dead) if you enable the option "surv": □ property □ cum ☑ surv

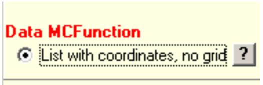
To provide an example we created the data set Book_Fig2_16a_sd.mcf which is identical to Book_Fig2_16a.mcf, except that the 200 points with the lowest value of the mark are defined to be dead (i.e., mark 0) and the others are defined to be surviving (i.e., mark 1).

Example *Book_Fig2_16a_sd.res*

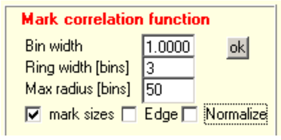
1. Select "**Mark correlation functions**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file "Book_Fig2_16a_sd.mcf"
3. Click "**List with coordinates, no grid**" in MCFunction
4. Provide in the window mark correlation functions the bin width in data units, an appropriate ring width, and a maximal distance r of the analysis. Select in the example a bin of 1m, a ring width of 3m and a maximal distance of 50m.
5. If you use a ring width of 1 unit, you can later use the function "**Combine replicates**" to load the results of the analysis, to change the ring width, and to use the corresponding cumulative summary function (see below "View results of mark correlation analysis").
6. Disable "**Normalize**" since you need non-normalized mark correlation functions and click button "**Calculate Index**".
7. Go to the window Select one test function that appears, de-select the small checkbox "cum" (to use the non-cumulative density correlation function) and check the small checkbox "surv" (because the mark is binary) at the bottom of the window □ cum ☑ surv. Now the non-cumulative "Density correlation function" can be selected after clicking again the button "**Calculate Index**"



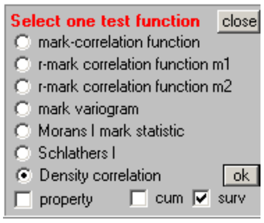
What do you want to do?
○ Standard analysis
● Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
□ Calculate simulation envelopes
Input data Expand update ok
Book_Fig2_16a.mcf
Book_Fig2_16a_sd.mcf
Book_Fig3_27.mcf
DataType8_Fig2_16a_sd.mcf
DataType9_triv2.mcf



Data MCFunction
● List with coordinates, no grid ?

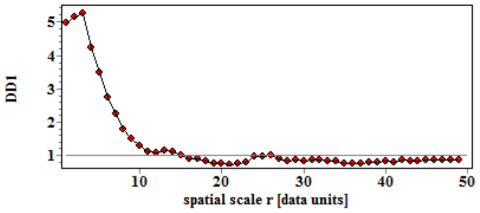


Mark correlation function
Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50
☑ mark sizes □ Edge □ Normalize

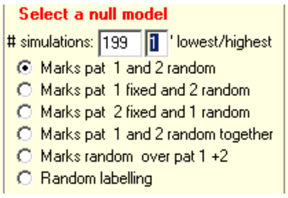


Select one test function close
○ mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
● Density correlation ok
□ property □ cum ☑ surv

8. The $dd_1(r)$ shows strong density effects up to distances of 10m with a peak at some 3m where dead individuals have, compared to surviving individuals, five time more neighbors:



9. Select an appropriate null model, the number of simulations of the null model (199), and the rule for the simulation envelopes (5’ lowest and highest).
10. Press button “**Calculate Index**” and *Programita* shows the observed and simulated pattern.

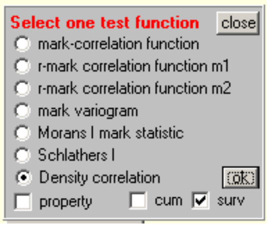


Select a null model

# simulations: 199 1 ' lowest/highest

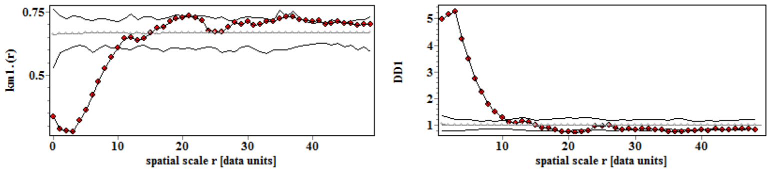
● Marks pat 1 and 2 random
○ Marks pat 1 fixed and 2 random
○ Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
○ Random labelling

11. Use the radio buttons of the window Select one test function to select a mark correlation function and click the small “**ok**” button to get the result graphic. As expected the *r*-mark correlation function shows that the survival of points that have another point within distance 10m is lower than expected. The $dd_1(r)$ shows strong density effects, dead individuals have, compared to surviving individuals, up to 5 time more neighbors within neighborhoods of 10m:

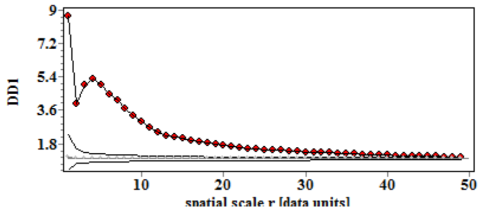


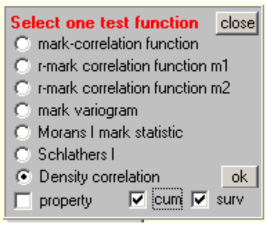
Select one test function — close

○ mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
● Density correlation — ok
☐ property ☐ cum ☑ surv



12. To obtain the cumulative $DD_1(r)$ function enable the small check box “cum” and repeat the analysis by clicking the button “**Calculate Index**”:





Select one test function — close

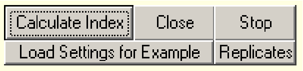
○ mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
● Density correlation — ok
☐ property ☑ cum ☑ surv

13. The cumulative $DD_1(r)$ is the density of points (i.e., s+d) within distance *r* around dead points divided by the density of points within distance *r* around surviving points, i.e., $DD_1(r) = K_{d,s+d}(r)/K_{s,s+d}(r)$.

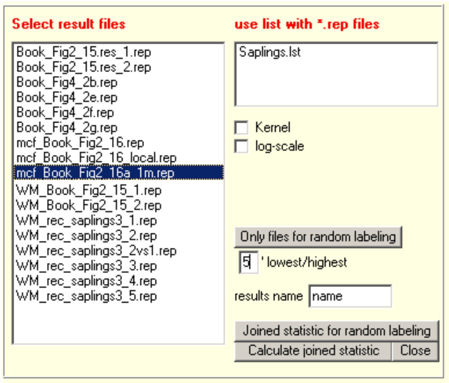
View results of mark correlation analysis

After conducting a mark correlation analysis you should save the results with button "**Save results**". *Programita* creates a results file names.res that also contains all settings of your analysis and an additional file name.rep which allows you to view and save the results for all mark correlation functions. Note that changing the ring width works only correctly if you selected a ring width of one unit for generating the data (i.e., you obtain non-overlapping rings).

You can access the procedure for loading the results with button "**Replicates**". If you conduct your analysis with a ring width of 1, the **Replicates** option allows you additionally to estimate the mark correlation function with different ring widths and to estimate the analogous cumulative mark correlation functions. To access this option follow the steps below:



1. select "**Replicates**"
2. highlight the *.rep results file you want to analyze in the window Select result files (mcf_Book_Fig2_16a_1m.rep)
3. Select the rule for the simulation envelopes (insert integer before **'lowest/highest)**. For example, if you conducted 199 simulations of the null model you may select 5 (i.e., the simulating envelopes are the 5th lowest and highest values).


Select result files

Book_Fig2_15.res_1.rep
Book_Fig2_15.res_2.rep
Book_Fig4_2b.rep
Book_Fig4_2e.rep
Book_Fig4_2f.rep
Book_Fig4_2g.rep
mcf_Book_Fig2_16.rep
mcf_Book_Fig2_16_local.rep
mcf_Book_Fig2_16a_1m.rep
WM_Book_Fig2_15_1.rep
WM_Book_Fig2_15_2.rep
WM_rec_saplings3_1.rep
WM_rec_saplings3_2.rep
WM_rec_saplings3_2vs1.rep
WM_rec_saplings3_3.rep
WM_rec_saplings3_4.rep
WM_rec_saplings3_5.rep

use list with *.rep files

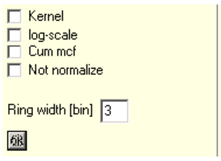
Saplings.lst

☐ Kernel
☐ log-scale

5 ' lowest/highest
results name: name

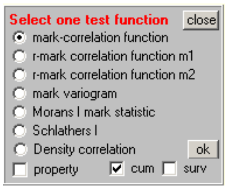
Only files for random labeling
Joined statistic for random labeling
Calculate joined statistic
Close

4. Click button "**Calculate joined statistic**" and *Programita* shows you the results of the analysis.
5. To change the ring width or to use the **cumulative mark correlation function** select the ring width after "**Ring width [bin]**" or enable the check box "**Cum mcf**" for the cumulative mark correlation function [see section 3.1.7.2 "Univariate Marked K-Functions (Data Type 6)" in Wiegand and Moloney 2014]. You can also plot the results on a logarithmic x-axis with check box "**log-scale**". (This can only be done if the ring width in the analysis was one). You can also show the non-normalized mark correlation functions ("**Not normalize**"). Finally, press the small **ok** button and *Programita* shows the results with the modified estimator.


☐ Kernel
☐ log-scale
☐ Cum mcf
☐ Not normalize

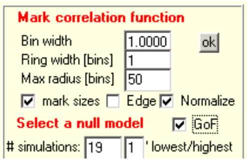
Ring width [bin] 3

6. In the window Select one test function you can view different mark correlation functions based on the modified estimator.


Select one test function

● mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation
☐ property ☑ cum ☐ surv

7. Using the button "**Save results**" you can save the results for this mark correlation function as *.res file.
8. You can also conduct the GoF test by checking the small box "**GoF**". To this end first select in the window that appears the distance interval of the GoF test (t0 and t1), the button "**Uni**" or "**Bi**" depending if the analysis is uni- or bivariate, and finally the button "**Calculate GoF rank**" to get the rank and the P-value of the test.


Mark correlation function

Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50
☑ mark sizes ☐ Edge ☑ Normalize

Select a null model ☑ GoF

simulations: 19 1 ' lowest/highest

6.2.3 Local independent marking, data type 6

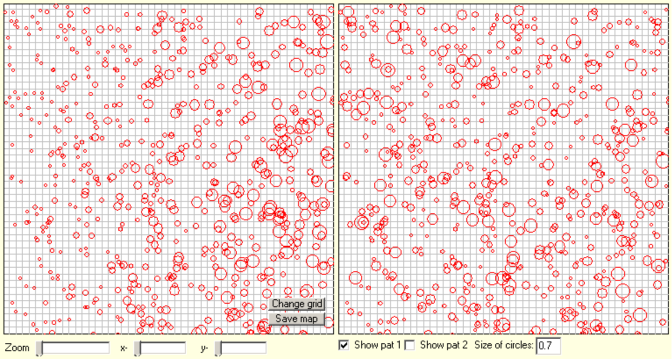
Mark correlation analysis may also be impacted by heterogeneity. In this case the value of the marks may be influenced by environmental covariates and we may observe systematic spatial trends in the values of the marks. For example, the size of the trees may be larger at the eastern part of an observation window than at the western part. This may cause larger-scale departures for example in the mark variogram.

The effect of a large-scale heterogeneity in the marking can be approximately factored out in the same way as for random labeling for qualitatively marked patterns. The marks in standard independent marking null model are shuffled in a way that the mark of each point can be exchanged with that of any other point in the entire observation window. However, localized independent marking exchanges only marks of points which are located closer than a given distance *R*. This removes the small-scale correlation structure in the marks, but maintains their observed large-scale correlation structure. Technically, all n_{1+2} points i of the marked pattern are numbered and the entries of the array *nr*[i] that runs from 1 to n_{1+2} are randomly permutated only if the coordinates of the point pair i – j are not farther away than distance *R*. In this way a given mark will normally not be moved more than distance *R* away from its original location.

Local independent marking, example CSR_grad_local.res

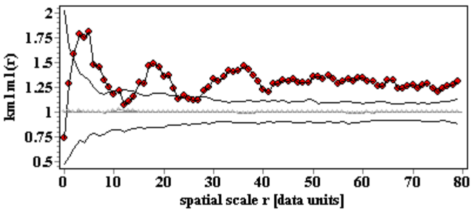
This example uses the data set CSR_grad.mcf that is based on a random pattern with random marks (and mean mark μ) within a 500m × 500m observation window, but afterwards the marks of the points were multiplied by factor x/500. Thus, the average size of the marks in dependence on the x-value is $m(x) = \mu\, x/500$ Thus, there is a systematic gradient in the marks from west to east where the marks become increasingly larger when moving eastward. However, except this gradient the marks do not show any spatial correlations.

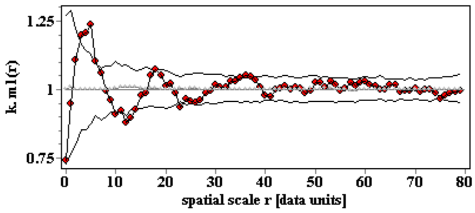
1. Select "**Mark correlation functions**" in window What do you want to do?
2. Highlight data file CSR_grad.mcf in Input data
3. Click "**List with coordinates, no grid**" in MCFunction
4. Press button "**Calculate Index**"
5. Select in mark correlation functions a bin width of 5, and a ring width of 1 and a maximal radius of 80.
6. Check "Calculate simulation envelopes" in window What do you want to do? and select "**Marks pat 1 and 2 random**". In a first step we use global independent marking. Select the number of simulations of the null model (199), and the rule for the simulation envelopes (5' lowest and highest). For "Size of circles" use a value of 0.7.
7. Press button "**Calculate Index**" and *Programita* shows the observed and simulated patterns. In the data the marks in the west are smaller than the marks in the east (left), but not in the null model (right):

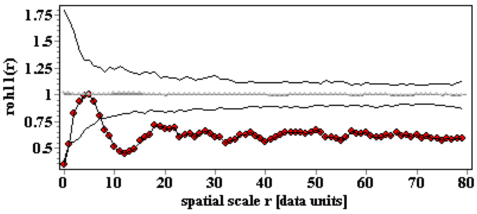


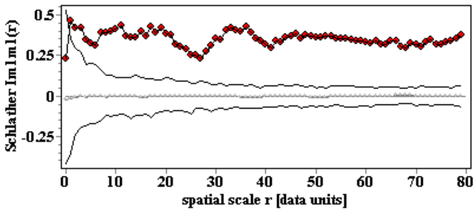
☑ Show pat 1 □ Show pat 2 Size of circles: 0.7

8. Use the radio buttons of the window Select one test function to select a mark correlation function and click the small "**ok**" button to get the result graphic:
9. Various of the mark correlation functions show larger scale departures from the independent marking null model:



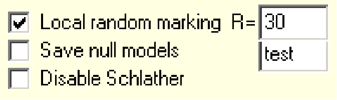






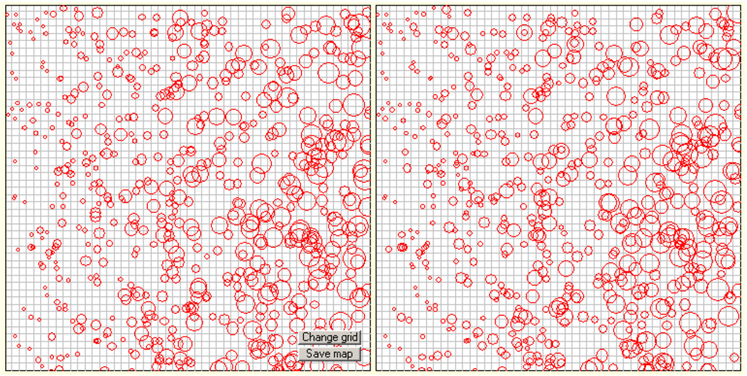
Because the average size of the marks depends linearly on their x-coordinate [i.e., $m(x) = \mu\, x/500$], the mark product of points separated by distance r will be larger than μ^2. Consequently, we observe a positive departure in the mark correlation function $k_{\mathrm{mm}}(r)$. However, the x-dependence in the mark is linear in the r-mark correlation functions and therefore averages out. The mark variogram shows negative departures because nearby marks are by construction similar in size, and the Moran's I type correlation coefficient $I_{\mathrm{mm}}(r)$ shows for the same reason positive departures. Especially, the marks are strongly correlated in x-direction.

10. To approximately factor out the large-scale heterogeneity click the option "**Local independent marking**" and select an appropriate maximal distance R for points that should switch their marks (select here R = 30).

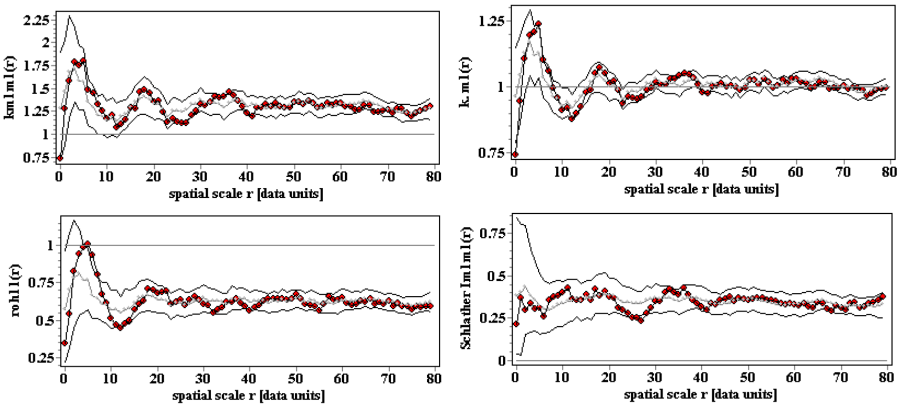


☑ Local random marking R= 30
□ Save null models test
□ Disable Schlather

11. Press button "**Calculate Index**" and *Programita* shows the observed and simulated patterns. The null model shuffles the marks only locally and the marks in the east are larger than in the west, now in the data and the null model:

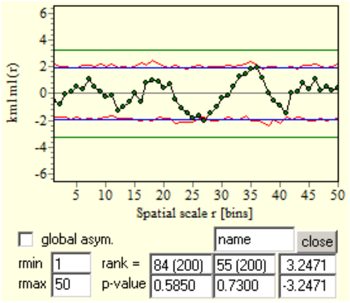


12. Use the radio buttons of the window Select one test function to select a mark correlation function and click the small "**ok**" button to get the result graphic. The expectation of the mark correlation functions under the local independent marking null model differ substantially from that of the independent marking null model and approximate the observed mark correlation functions. This is because the marks do not show other correlations than that imposed by the gradient.



13. Using the GoF test you can verify that there is indeed no departure from the local independent marking null model:

$k_{mm}(r)$

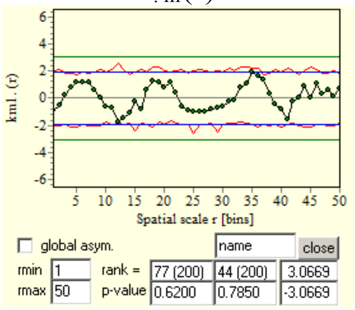


☐ global asym. name

rmin 1 rank = 84 (200) 55 (200) 3.2471

rmax 50 p-value 0.5850 0.7300 -3.2471

$k_{.m}(r)$

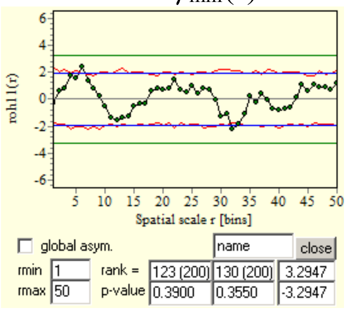


☐ global asym. name

rmin 1 rank = 77 (200) 44 (200) 3.0669

rmax 50 p-value 0.6200 0.7850 -3.0669

$\gamma_{mm}(r)$

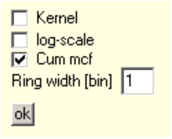


☐ global asym. name

rmin 1 rank = 123 (200) 130 (200) 3.2947

rmax 50 p-value 0.3900 0.3550 -3.2947

If you conducted the analysis with a ring width of one unit and use the "**Combine replicate**" option (files mcf_Book_Fig2_16a_grad_1m.rep, Book_Fig2_16a_grad_1m.res) to view the results of the mark correlation analysis you can also use the cumulative mark correlation functions [see section 3.1.7.2 "Univariate Marked K-Functions (Data Type 6)" in Wiegand and Moloney 2014]. Enable the checkbox "**Cum mcf**" to obtain the cumulative functions:

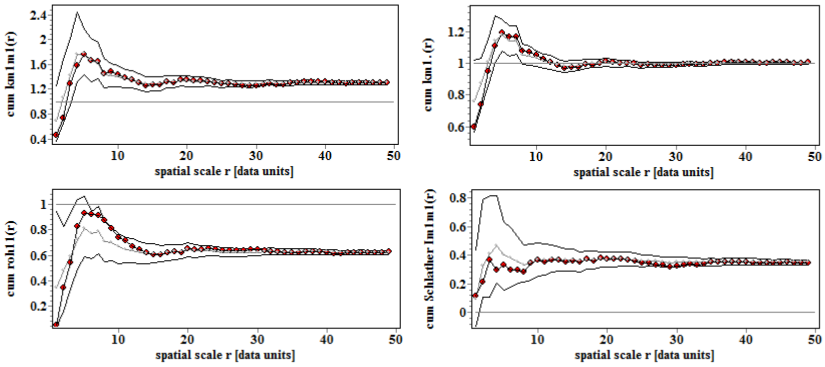


☐ Kernel

☐ log-scale

☑ Cum mcf

Ring width [bin] 1



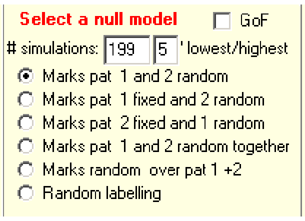
6.3 Analysis of univariate patterns with two marks (data type 7)

Quantitatively marked patterns of this type comprise the coordinates of a univariate pattern and **each point carries two additional quantitative marks** that characterize the ecological object that is idealized as point. The quantitative marks are usually a continuous attribute such as the size and height of a tree, but can also be an integer number such as the number of orchids of two species located on host trees, or the number of seeds of two species in feces of seed dispersing animals. Because the bivariate mark variogram is sensitive to differences in the means μ_1 and μ_2 of the two marks you should normalize the marks to yield the same mean. (The other mark correlation functions are independent on the absolute values.)

The major interest in analysis of quantitatively marked patterns with two quantitative marks is to find out whether the two marks show some spatial correlation that depends on the distance *r* between points, conditional on the underlying univariate pattern. In a way this is similar to testing for independence between the two component patterns of a bivariate pattern. For example, the two orchid species may tend to be placed less frequently together on nearby host trees than expected by independent placement, or the seeds of the two species tend to be more frequently placed together in nearby feces than expected.

Depending on the ecological question two types of null models are possible; see section 3.1.7.3 “Two Quantitative Marks Attached to a Univariate Pattern (Date Type 7)” in Wiegand and Moloney (2014):

Null model type 1
For example, if the marks are the number of orchids of two species, we may ask whether they are independently distributed over the host trees. In this case, we can condition on the number of orchids of the first species and shuffle only the second mark (i.e., number of individuals of the second orchid species) randomly over the trees of the univariate pattern (**Marks of pat 1 fixed and 2 random**). If none of the two orchid species is antecedent we can also condition on the number of orchids of the second species and shuffle the first mark (**Marks of pat 2 fixed and 1 random**). If appropriate, we may also randomize the locations of both marks (**Marks of pat 1 and 2 random**). This null model thus tests if the placement of the two orchid species on the host trees was spatially independent as opposed by positive or negative associations that could be promoted by species interactions between the two species or by shared or opposed habitat requirements.


Select a null model □ GoF
# simulations: 199 5 ' lowest/highest
● Marks pat 1 and 2 random
○ Marks pat 1 fixed and 2 random
○ Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
○ Random labelling

Null model type 2
If we analyze the spatial correlation in marks representing the number of seeds of two species in feces (e.g., Fedriani et al. 2014), we ask if there is a spatial correlation in the co-occurrence of the two marks (i.e., number of seeds of the two species). In this case we cannot separate the two marks because they occurred together in the same feces. Thus, we need to shuffle the vector of marks of the point i, given by (m_{i1}, m_{i2}), randomly over the points of the univariate pattern (**Marks pat 1 and 2 random together**). Similar augments can be made for example for the case where the marks are the size and the height of a tree. Here the null model simulation must also keep these two properties together.

6.3.1 Data preparation for data type 7

The data files must be an ASCII file with the *.mcf extension with the following format :

```
0  200  0  191  600
 51.14   48.99  12  0.80  0.59
112.80   45.85  12  1.35  1.08
  5.59   61.90  12  0.42  1.73
  6.41   62.55  12  0.61  1.57
 53.95   74.64  12  0.83  0.58
123.54    4.92  12  1.06  1.02
…
```

where the first line gives the size of the observation window (200 × 191 units in the example) and the number of points in the pattern.

- the first two columns of the following lines are the coordinates of the points. The third column must have the value “12” for all points which indicates that each point can carry two marks.
- The forth column carries the first mark of the point
- The fifth column carries the second mark of the point.

The data file must be a space or tab delimited ASCII file with the *.mcf extension.

6.3.2 Steps data type 7 (DataType7.res)

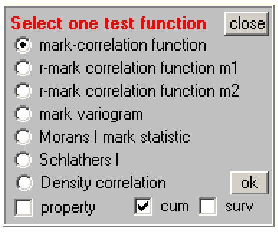
Programita estimates for the first mark the univariate mark correlation functions described above, and additionally the corresponding bivariate test statistics that consider the two quantitative marks.

To understand the bivariate mark correlation functions it is important to note that they involve the **first mark m_{i1} of the first point i and the second mark m_{j2} of the second point j**. A bivariate mark correlation function for this data type therefore estimates the mean value $c_t(r)$ of a test function $t(m_{i1}, m_{j2})$ taken over all pairs of points i and j that are located at distance r, and normalizes with the mean c_t taken over all pairs of points.

For example, the bivariate mark variogram $\gamma_{m1m2}(r)$ estimates the squared difference $0.5(m_{i1} - m_{j2})^2$ between the first mark m_{i1} of the first point i and the second mark m_{j2} of the second point j which is located at distance r of the first point. Note that the bivariate r-mark correlation functions $k_{m1\,.}(r)$ and $k_{.\,m2}(r)$ are the same as the univariate r-mark correlation functions for marks 1 and 2, respectively, and therefore not of interest for a bivariate analysis. Also, the density correlation functions are not relevant for data type 7 because they do not differ from the corresponding univariate functions.

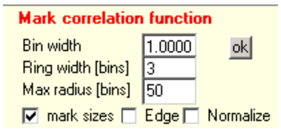
The bivariate methods are explained in detail in section 3.1.7.3 “Two Quantitative Marks Attached to a Univariate Pattern (Date Type 7)” in Wiegand and Moloney (2014):

- the mark correlation function $k_{m1m2}(r)$
- the r-mark correlation function $k_{m1\,.}(r)$
- the r-mark correlation function $k_{.\,m2}(r)$
- the mark variogram $\gamma_{m1m2}(r)$
- a Moran’s I type mark statistics $I_{m1m2}(r)$
- Schlather’s correlation function $I_{m1m2}(r)$


Select one test function
close
● mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation
ok
□ property ☑ cum □ surv

Note that the two correlation coefficients $I_{m1m2}(r)$ may show absolute values larger than one if the number of point pairs at this distance is very low (say < 10). To avoid this increase the ring width.

The default for mark correlation functions are the normalized functions, i.e., $k_t(r) = c_t(r)/c_t$ where c_t is the normalization constant for a given test function t. However, you can also use the non-normalized mark correlation functions $c_t(r)$ when disabling the checkbox "**Normalize**".



Mark correlation function

Bin width 1.0000 ok

Ring width [bins] 3

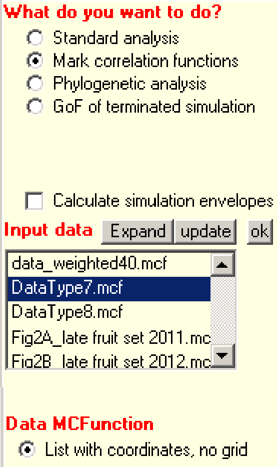
Max radius [bins] 50

☑ mark sizes □ Edge □ Normalize

The mark correlation mode can be accessed with the following sequence of actions:

1. Select "**Mark correlation functions**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file "DataType7.mcf".

   This data file was generated by using an intensity function $\lambda(\mathbf{x})$ (int_Book_Fig2_26_R1_30.int) and simulating the two marks $m_1(\mathbf{x})$ and $m_2(\mathbf{x})$ for 600 random points (i.e., following CSR) stochastically with $m(\mathbf{x}) = \lambda(\mathbf{x})\ (0.3 + \varepsilon\ ) + \varepsilon$ where ε is a random number equally distributed between 0 and 1. As a consequence the values of the marks are positively correlated. These deterministic relationships are made noisy with the factor ε.



What do you want to do?

○ Standard analysis

● Mark correlation functions

○ Phylogenetic analysis

○ GoF of terminated simulation

□ Calculate simulation envelopes

Input data Expand update ok

data_weighted40.mcf

DataType7.mcf

DataType8.mcf

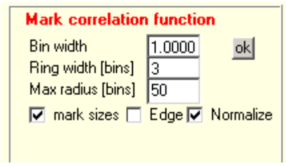
Fig2A_late fruit set 2011.mc

Fig2B_late fruit set 2012.mc

Data MCFunction

● List with coordinates, no grid

3. Click "**List with coordinates, no grid**" in MCFunction
4. Provide in the window mark correlation functions the bin width in data units, an appropriate ring width, and a maximal distance *r* of the analysis. Select in the example a bin of 1m, and a ring width of 3m.
5. If you use a ring width of 1 unit, you can later use the function "**Combine replicates**" to load the results of the analysis, to change the ring width, and to use the corresponding cumulative summary function (see "View results of mark correlation analysis").
6. Disable "**Normalize**" if you want to use the non-normalized mark correlation functions. The default is "**Normalize**"
7. Check "**Edge**" if you want to use the Ripley edge correction. Default is no edge correction. Note that edge correction is not required for mark correlation functions.
8. Check "Calculate simulation envelopes" in window What do you want to do? The subwindow "Select a null model" appears.



Mark correlation function

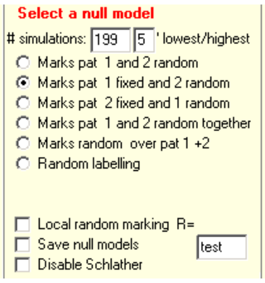
Bin width 1.0000 ok

Ring width [bins] 3

Max radius [bins] 50

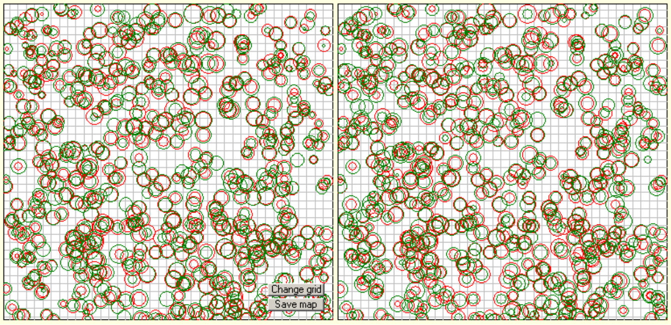
☑ mark sizes □ Edge ☑ Normalize

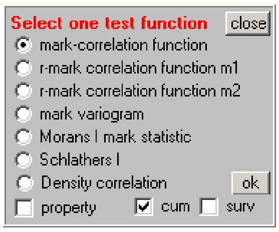
9. Select an appropriate null model, the number of simulations of the null model (199), and the rule for the simulation envelopes (5' lowest and highest). Note that not all null models are appropriate for all mark correlation function data types and that they must be selected with care.
   Select for the null model "**Marks pat 1 fixed and 2 random**" that shuffles the first mark randomly over the points. (You can also use the alternative null model "**Marks pat 2 fixed and 1 random**", for example if you want to use the density correlation function).



Select a null model
# simulations: 199 5 ' lowest/highest
○ Marks pat 1 and 2 random
● Marks pat 1 fixed and 2 random
○ Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
○ Random labelling
☐ Local random marking R=
☐ Save null models test
☐ Disable Schlather

10. Press button "**Calculate Index**" and *Programita* shows the observed and simulated pattern. The red circles represent the first mark and the green circles the second mark, and the area of the disk is proportional to the mark:



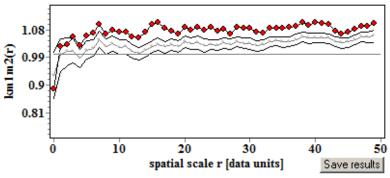


Select one test function close
● mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation ok
☐ property ☑ cum ☐ surv

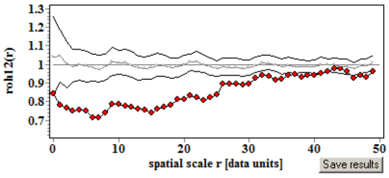
You notice that the points do not change their location, but that the size of the green points changes because the second mark is randomly shuffled over all points.

11. Use the radio buttons of the window Select one test function to select a mark correlation function and click the small "**ok**" button to get the result graphic:

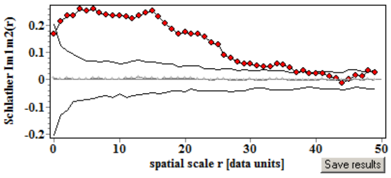
$k_{\mathrm{m1m2}}(r)$



$\gamma_{\mathrm{m1m2}}(r)$



$I_{\mathrm{m1m2}}(r)$



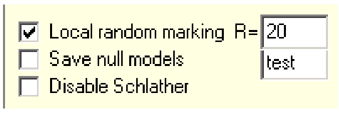
The *r*-mark correlation functions the density correlation functions of data type 7 are not of interest here because they correspond to the univariate functions.

19. As expected by the construction of the pattern, the bivariate mark correlation function $k_{\mathrm{m1m2}}(r)$ shows that the product of the first mark m_{i1} of the first point i and the second mark m_{j2} of the second point j is larger than expected by the null model. Nearby points have marks m_{i1} and m_{j2} that are more similar than expected (mark variogram) and the values of m_{i1} and m_{j2} are positively correlated (Schlather's I).

Local independent marking for univariate patterns with two marks

The pattern DataType7.mcf was generated without simulating spatial interactions between the two marks and the larger-scale spatial correlations in the two marks were only imposed by the intensity function. To show this we can use the null model variant that conducts the randomization of the marks only locally.

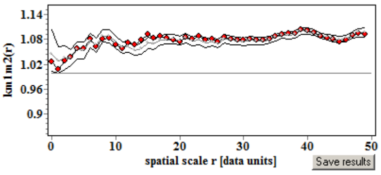
1. Read the settings of the previous example (*Example DataType7.res*) using the "Load Settings for Example" button, but use a ring width of 5.
2. To approximately factor out the large-scale heterogeneity in the marks click additionally the option "**Local independent marking**" and select an appropriate maximal distance R for points that should switch their marks (select here R = 20).



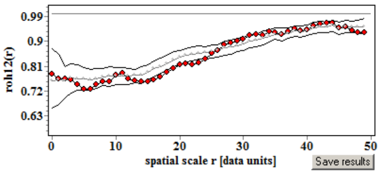
☑ Local random marking R= 20
□ Save null models test
□ Disable Schlather

3. Press button "**Calculate Index**" and *Programita* shows the observed and simulated pattern. The red circles represent the first mark and the green circles the second mark, and the area of the disk is proportionally to the mark.
4. The mark correlation functions are now within the pointwise simulation envelopes except for a weak departure for the $I_{\mathrm{m1m2}}(r)$:

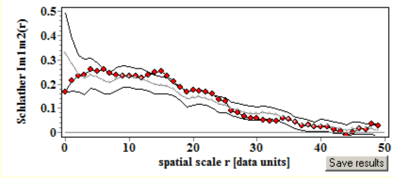
$k_{\mathrm{m1m2}}(r)$



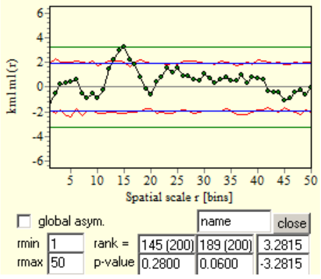
$\gamma_{\mathrm{m1m2}}(r)$



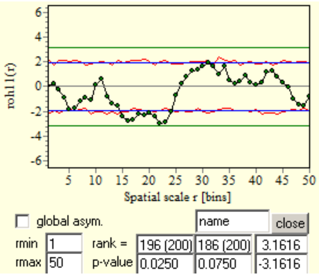
$I_{\mathrm{m1m2}}(r)$



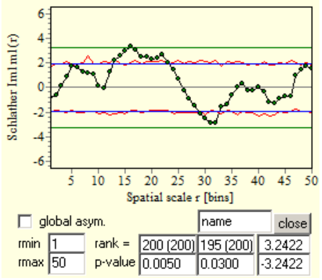
which is confirmed by the global envelope tests:



□ global asym. name
rmin 1 rank = 145 (200) 189 (200) 3.2815
rmax 50 p-value 0.2800 0.0600 -3.2815



□ global asym. name
rmin 1 rank = 196 (200) 186 (200) 3.1616
rmax 50 p-value 0.0250 0.0750 -3.1616



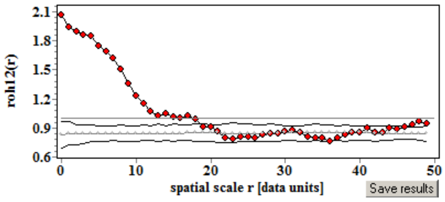
□ global asym. name
rmin 1 rank = 200 (200) 195 (200) 3.2422
rmax 50 p-value 0.0050 0.0300 -3.2422

Independent marking of mark vector for univariate patterns with two marks

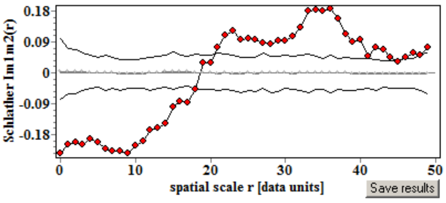
If the two marks of a point constitute a unit such as seeds of different species that were dispersed together in the same feces or if the marks characterize two different properties of the point (e.g., height and dbh), the randomization of the null model needs to shuffle the entire vector of marks (m_{i1}, m_{i2}) over the points i of the pattern. In the example Book_Fig2_16_bi.res the first mark attached to a point is proportional to one over the number of neighbors within 10 m and the second mark is the number of points within 10m.

1. Select "**Mark correlation functions**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file "Book_Fig2_16_bi.mcf".
3. Click "**List with coordinates, no grid**" in MCFunction
4. Provide in the window mark correlation functions the bin width in data units, an appropriate ring width, and a maximal distance *r* of the analysis. Select in the example a bin of 1m, and a ring width of 5m.
5. Check "Calculate simulation envelopes" in window What do you want to do? The subwindow "Select a null model" appears.
6. Select for the null model "**Marks pat 1 and 2 random together**" that shuffles the vector (m_{i1}, m_{i2}) of marks randomly over the points, the number of simulations of the null model (199), and the rule for the simulation envelopes (5' lowest and highest).
7. Press button "**Calculate Index**" and *Programita* shows the observed and simulated patterns. The red circles represent the first mark and the green circles the second mark, and the area of the disk is proportional to the mark.
8. The bivariate mark variogram shows that the two marks of nearby points are up to distances of 20m more different than expected and bivariate Schlather's I indicates for distances up to 19m a negative correlation between the two marks of nearby points (and between 23 and 40m a positive correlation):

$\gamma_{\mathrm{m1m2}}(r)$



$I_{\mathrm{m1m2}}(r)$



Clearly, this is caused by the construction of the marks, the first is proportional to the inverse of the number of points within 10m and the second is the number of points within 10m. Interestingly, the specific null model that keeps the marks m_1 and m_2 of a point together produces for the mark variogram Ubut not for the other mark correlation functions) an expectation different from the random value of one. Thus, the bivariate mark variogram is especially sensitive to the correct assignment of the null model, depending on the data type.

6.4 Pattern with one qualitative and one quantitative mark (data type 8)

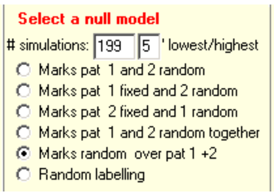
Programita allows also analyzing a data type where each point contains one qualitative and one quantitative mark. For example, the quantitative mark could be the size of a tree and the qualitative mark indicates whether the tree survived or died during the last 5 years. If for example surviving is coded as pattern 1 and dead as pattern 2, and the quantitative mark is size, the univariate mark correlation functions describe the relationships between the quantitative mark m_i of surviving focal trees i and the quantitative mark m_j of another surviving trees j distance *r* apart. In contrast, the bivariate mark correlation functions describe the relationships between the quantitative mark m_i of surviving focal trees i and the quantitative mark m_j of dead trees j distance *r* apart.

There are two types of null models possible for this data structure, one may randomize the quantitative mark or one may randomize the qualitative mark. The decision between the two null models depends on the data and the ecological question on hand. Details on this data type are provided in see section 3.1.7.4 "One Qualitative and One Quantitative Mark (Data Type 8)" in Wiegand and Moloney (2014).

However, because this data structure looks at two different marks of the same individual, its applicability may be somewhat reduced. The most useful null model is probably that of random labeling that explores if the two types of points (defined by the qualitative mark) differ in the spatial structure of their quantitative marks.

Null model type 1: randomize quantitative mark

This null model will most likely detect a systematic difference in the quantitative mark (e.g., size) between the two types of points (e.g., surviving vs. dead). The null model fixes the qualitative marks (e.g., surviving and dead), but shuffle the quantitative mark is randomly shuffled over all points (e.g., surviving and dead trees) (**Marks random over pat 1 + 2**).



Select a null model

# simulations: 199 5 ' lowest/highest

○ Marks pat 1 and 2 random

○ Marks pat 1 fixed and 2 random

○ Marks pat 2 fixed and 1 random

○ Marks pat 1 and 2 random together

● Marks random over pat 1 +2

○ Random labelling

Null model type 2

Alternatively, the "random labeling" null model randomly shuffles the qualitative mark (e.g., surviving vs. dead) over all locations of the points but keeps the quantitative mark (e.g., size) as fixed **(Random labeling**).

6.4.1 Data preparation for data type 8

The data files must be an ASCII file with the *.mcf extension with the following format (the example are the first lines of file DataType8.mcf):

```
0  500  0  500  600
0.60   35.35  1  1.281  1.281
0.70  274.90  1  1.976  1.976
1.80  274.60  2  2.138  2.138
1.15  342.20  2  3.072  3.072
1.10  385.85  1  1.361  1.361
…
```

where the first line gives the size of the observation window (500 × 500 units in the example) and the number of points in the pattern.

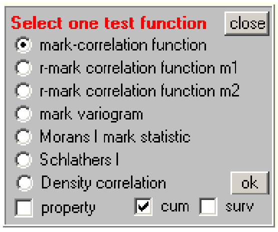
- the first two columns of the following lines are the coordinates of the points. The third column codes the qualitative mark; here a “1” for dead and a “2” for surviving.
- the fourth and fifth column carry the mark of the type 1 and 2 point, respectively (you can write the mark of the points in both columns).

6.4.2 Steps for data type 8 (DataType8.res)

Programita estimates for the univariate pattern of type 1 points the univariate mark correlation functions and the corresponding bivariate test statistics considering the marks of the two types of points. However, because the two types of null models randomize the marks over the entire pattern, the results of the univariate analysis do not coincide with the results of the univariate analysis of the type 1 points (i.e., data type 6).

To understand the bivariate mark correlation functions of data type 8 it is important to note that the bivariate mark correlation function estimates for this data type the mean value $c_t(r)$ of a test function $t(m_i, m_j)$ taken over all type 1 – type 2 pairs of points i and j distance r apart. This conditional mean is then normalized with the mean c_t taken over all type 1 – type 2 pairs of points i and j, irrespectively of their distance.

The bivariate methods are explained in detail in section 3.1.7.4 “One Qualitative and One Quantitative Mark (Data Type 8)” in Wiegand and Moloney (2014). You can select the following “uni” and bivariate mark correlation functions:


Select one test function
● mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation
□ property ☑ cum □ surv

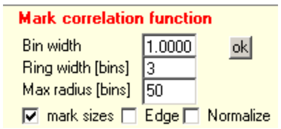
- the mark correlation function $k_{\mathrm{mm}}(r)$
- the r-mark correlation function $k_{\mathrm{m}\,.}\,(r)$
- the r-mark correlation function $k_{\,.\,\mathrm{m}}(r)$
- the mark variogram $\gamma_{\mathrm{mm}}(r)$
- a Moran’s I type mark statistics $I_{\mathrm{mm}}(r)$
- Schlather’s correlation function $I_{\mathrm{mm}}(r)$
- the density correlation functions $C_{m,g}(r)$ or $C_{m,K}(r)$

Note that the two correlation coefficients $I_{\mathrm{mm}}(r)$ may show absolute values larger than one if the number of point pairs at this distance is very low (say < 10). To avoid this increase the ring width.

Because the two null models randomize over all points (either shuffling the qualitative or the quantitative mark over all points), both the "univariate" and the bivariate mark correlation functions are of interest here. For example:

- the "univariate" r-mark correlation function $k_{\mathrm{m}\,.}(r)$ estimates the mean size of dead trees (type 1) of all dead-dead (i.e., type 1 – type 1) pairs at distance r whereas the
- the "bivariate" r-mark correlation function $k_{\mathrm{m}\,.}(r)$ estimates the mean size of dead trees (type 1) of all dead-surviving (i.e., type 1 – type 2) pairs at distance r
- the "univariate" mark variogram $\gamma_{\mathrm{mm}}(r)$ estimates the mean squared size difference between a dead tree (type 1) and another dead tree (type 1) at distance r whereas the
- the "bivariate" mark variogram $\gamma_{\mathrm{mm}}(r)$ estimates the mean squared size difference between a dead tree (type 1) and surviving tree (type 2) at distance r
- the "univariate" density correlation functions estimates the correlation between the size of dead trees (i.e., type1) and the density of dead trees (type 1) within distance r.
- the "bivariate" density correlation functions estimates the correlation between the size of dead trees (i.e., type1) and the density of surviving trees (type 2) within distance r.

The default for mark correlation functions are the normalized functions, i.e., $k_{\mathrm{t}}(r) = c_{\mathrm{t}}(r)/c_{\mathrm{t}}$ where c_{t} is the normalization constant for a given test function t. However, you can also use the non-normalized mark correlation functions $c_{\mathrm{t}}(r)$ when disabling the checkbox "**Normalize**".



Mark correlation function

Bin width 1.0000 ok

Ring width [bins] 3

Max radius [bins] 50

☑ mark sizes ☐ Edge ☐ Normalize

Random labeling for patterns with one qualitative and one quantitative mark

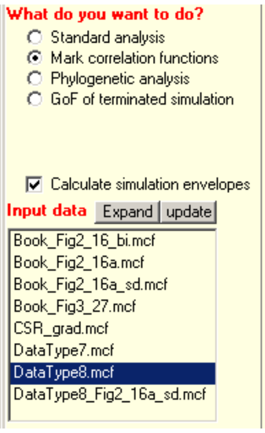
Data structure 8 has two possible null models, one randomizes the quantitative mark over all points (independent marking) and the other randomizes the qualitative mark over all points (i.e., random labeling).

Because the independent marking null model randomizes the quantitative mark over all points the normalization constants need to use as described in Wiegand and Moloney (214: section 3.1.7.4); i.e., the mean and standard deviation of the marks of the joined pattern.

However, the random labeling null model continuously changes what is pattern 1 and pattern 2. Therefore the test functions and the normalization constants must be the same as for data type 9; i.e., the means of the marks of the individual patterns and the covariance of pattern 1 and pattern 2. In this case the mean and the covariance must be used re-estimated for each simulation of the null model.

The mark correlation mode for a pattern with one qualitative and one quantitative mark can be accessed with the following sequence of actions:

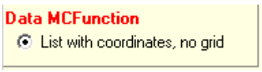
1. Select “**Mark correlation functions**” in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file “DataType8.mcf”.
   This data file was generated by using the data file Book_Fig2_16a.mcf and randomly assigning to 213 of the 600 points the type 2 and to the other 378 points the type 1. Thus, the null model “**Random labeling**” should not yield significant departures in none of the mark correlation functions.



What do you want to do?
○ Standard analysis
● Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ Calculate simulation envelopes
Input data

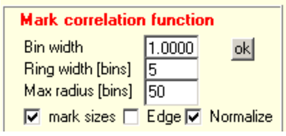
Book_Fig2_16_bi.mcf
Book_Fig2_16a.mcf
Book_Fig2_16a_sd.mcf
Book_Fig3_27.mcf
CSR_grad.mcf
DataType7.mcf
DataType8.mcf
DataType8_Fig2_16a_sd.mcf

3. Click “**List with coordinates, no grid**” in MCFunction



Data MCFunction
● List with coordinates, no grid

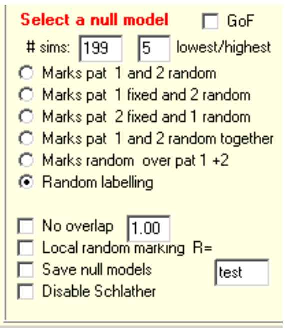
4. Provide in the window mark correlation functions the bin width in data units, an appropriate ring width, and a maximal distance *r* of the analysis. Select in the example a bin of 1m, and a ring width of 5m.



Mark correlation function
Bin width 1.0000
Ring width [bins] 5
Max radius [bins] 50

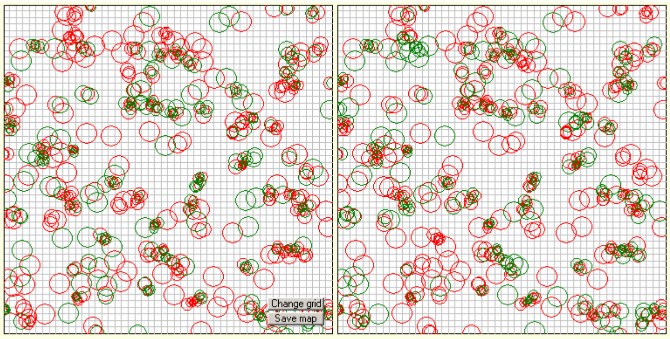
☑ mark sizes ☐ Edge ☑ Normalize

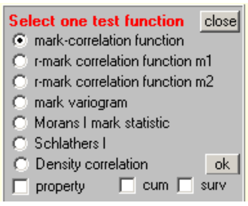
5. If you use a ring width of 1 unit, you can later use the function “**Combine replicates**” to load the results of the analysis, to change the ring width, and to use the corresponding cumulative summary function (see “View results of mark correlation analysis”).
6. Disable “**Normalize**” if you want to use the non-normalized mark correlation functions. The default is “Normalize”. Note that the estimators for the mark correlation function are that of data type 9.
7. Check “**Edge**” if you want to use the Ripley edge correction. Default is no edge correction. Note that edge correction is not required for mark correlation functions.
8. Check “**Calculate simulation envelopes**” in window What do you want to do? The subwindow “Select a null model” appears.
9. Select the number of simulations of the null model (199), and the rule for the simulation envelopes (5’ lowest and highest). Note that not all null models are appropriate for all mark correlation function data types and that they must be selected with care.
   Select the null model “**Random labeling**” that shuffles the qualitative mark randomly over the points.



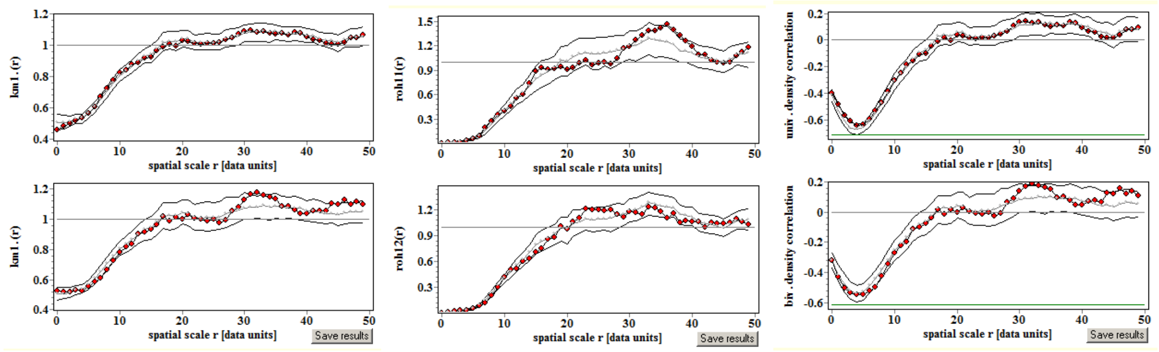
Select a null model ☐ GoF
# sims: 199 5 lowest/highest
○ Marks pat 1 and 2 random
○ Marks pat 1 fixed and 2 random
○ Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
● Random labelling
☐ No overlap 1.00
☐ Local random marking R=
☐ Save null models test
☐ Disable Schlather

   Press button “**Calculate Index**” and *Programita* shows the observed and simulated pattern. The red disks are type 1 points and the green disks are type 2 points, and the area of a disk is proportional to the mark. Note that the null model changes only the type (indicated by changing the red and green color), but not the location or size of the marks:




Select one test function | close
● mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation | ok
□ property □ cum □ surv

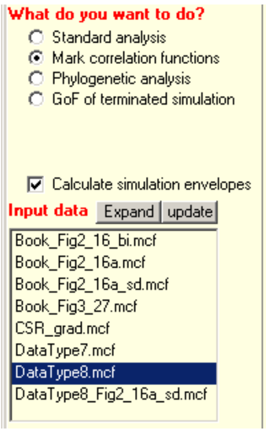
10. Use the radio buttons of the window Select one test function to select a mark correlation function and click the small "**ok**" button to get the result graphic:
11. As expected, all mark correlation functions are within the pointwise simulation envelopes of the random labeling null model:



12. Note that the null model of random labeling yields the univariate mark correlation functions of the joined pattern of type 1 and type 2 points as expectation. This clearly differs here from the expectation of independent marking where the values of the marks are randomly shuffled (see example below).

Independent marking for patterns with one qualitative and one quantitative mark.

We now apply the null model of independent marking (**Marks random over pat 1 + 2**) to the previous example.

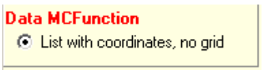

What do you want to do?
○ Standard analysis
● Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation

☑ Calculate simulation envelopes

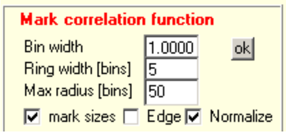
Input data | Expand | update
Book_Fig2_16_bi.mcf
Book_Fig2_16a.mcf
Book_Fig2_16a_sd.mcf
Book_Fig3_27.mcf
CSR_grad.mcf
DataType7.mcf
DataType8.mcf
DataType8_Fig2_16a_sd.mcf

1. Select "**Mark correlation functions**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file "DataType8.mcf".

   This data file was generated by using the data file Book_Fig2_16a.mcf and randomly assigning to 213 of the 600 points the type 2 and to the other 378 points the type 1. Thus, the null model "**Marks random over pat 1 + 2**" should yield significant departures if the quantitative marks show a spatial structure (as the case in Book_Fig2_16a.mcf).
3. Click "**List with coordinates, no grid**" in MCFunction


Data MCFunction
● List with coordinates, no grid

4. Provide in the window mark correlation functions the bin width in data units (1), an appropriate ring width (5), and a maximal distance *r* of the analysis (50).



Mark correlation function

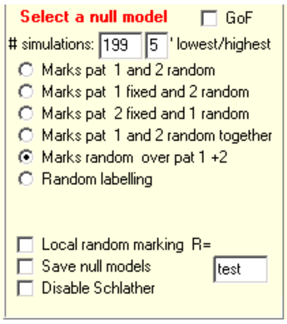
Bin width 1.0000

Ring width [bins] 5

Max radius [bins] 50

☑ mark sizes ☐ Edge ☑ Normalize

5. Check "**Calculate simulation envelopes**" in window What do you want to do? The subwindow "Select a null model" appears.
6. Select the null model "**Marks random over pat 1 + 2**" that shuffles the quantitative mark randomly over the points, the number of simulations of the null model (199), and the rule for the simulation envelopes (5' lowest and highest).



Select a null model ☐ GoF

simulations: 199 5 ' lowest/highest

○ Marks pat 1 and 2 random

○ Marks pat 1 fixed and 2 random

○ Marks pat 2 fixed and 1 random

○ Marks pat 1 and 2 random together

● Marks random over pat 1 +2

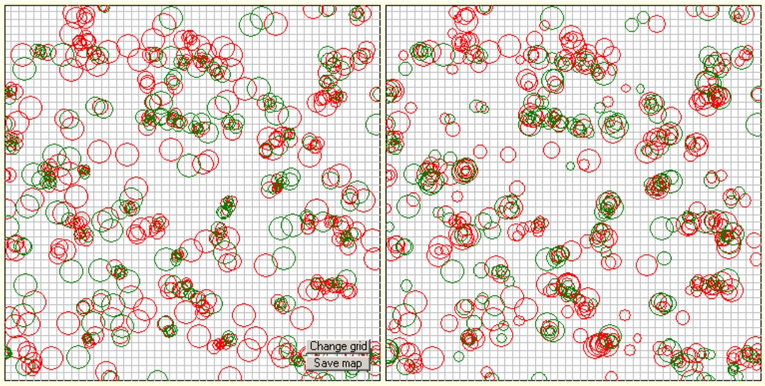
○ Random labelling

☐ Local random marking R=

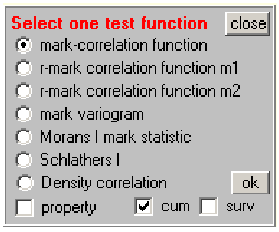
☐ Save null models test

☐ Disable Schlather

7. Press button "**Calculate Index**" and *Programita* shows the observed and simulated patterns. The red disks are type 1 points and the green disks are type 2 points, and the area of a disk is proportional to the mark. Note that the null model changes only the size, but not the location or type of the points (i.e., the color):



8. Use the radio buttons of the window Select one test function to select a mark correlation function and click the small "**ok**" button to get the result graphic: as expected, the mark correlation functions show departures from the independent marking null model that were caused by the small-scale correlations in the quantitative marks:



Select one test function

● mark-correlation function

○ r-mark correlation function m1

○ r-mark correlation function m2

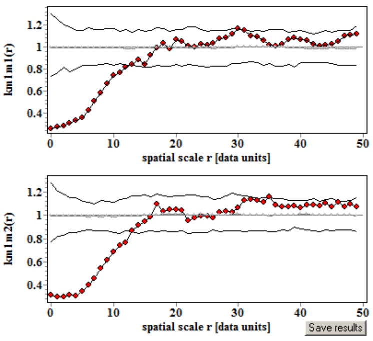
○ mark variogram

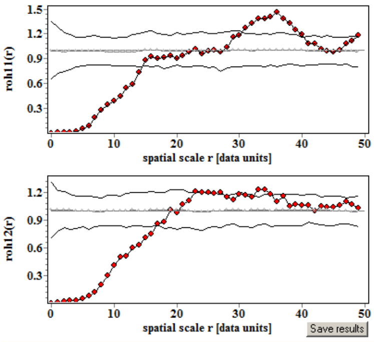
○ Morans I mark statistic

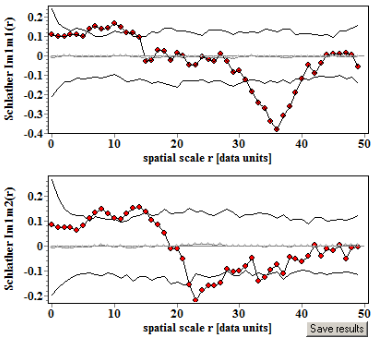
○ Schlathers I

○ Density correlation

☐ property ☑ cum ☐ surv







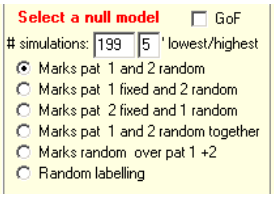
9. The independent marking null model can also be applied with local independent marking.

The two examples show that the selection between random labeling and independent marking null models depends on the underlying hypothesis and that the expectations of the two null models may greatly vary.

6.5 Bivariate patterns with one quantitative mark (data type 9)

Programita allows also analyzing a data type where a bivariate pattern carries one quantitative mark. For example, we may have two different tree species and the quantitative mark is the size of the trees. This data structure is of special interest to detect interactions between species.

The basic interest in analyzing patterns of this type is to explore the impact of proximity (and mark) of the first pattern (i.e., pattern 1) on the marking of the second pattern (i.e., pattern 2). The mark correlation functions for data type 9 are the same as for data type 8 with random labeling; however, the null models are fundamentally different. While the two alternative null models of data type 8 randomized the (qualitative or quantitative) marks over all points of the underlying univariate pattern, the null models of data type 9 must randomize the quantitative mark only within patterns. For example, we may keep the marks of the "antecedent" pattern 1 fixed and randomize only the marks of the second pattern. The normalization constants c_t are the same as that of data type 8 for random labeling. You can use three null models, one that randomizes the quantitative marks of both component patterns but only inside the patterns not across patterns (**Marks 1 and 2 random**), one that randomizes only the quantitative marks of pattern 2 (**Marks 1 fixed and 2 random**), and one that randomizes only the quantitative marks of pattern 1 (**Marks 2 fixed and 1 random**).



Select a null model □ GoF

simulations: 199 5 ' lowest/highest

● Marks pat 1 and 2 random
○ Marks pat 1 fixed and 2 random
○ Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
○ Random labelling

6.5.1 Data preparation for data type 9

The data files must be an ASCII file with the *.mcf extension with the following format (the same as for data type 8):

```
0   500   0   500   600
0.60   35.35  1   1.281   1.281
0.70  274.90  1   1.976   1.976
1.80  274.60  2   2.138   2.138
1.15  342.20  2   3.072   3.072
1.10  385.85  1   1.361   1.361
…
```

where the first line gives the size of the observation window (500 × 500 units in the example) and the number of points in the pattern.

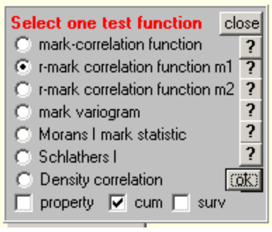
- the first two columns of the following lines are the coordinates of the points. The third column codes the component pattern of the underlying bivariate pattern; here a "1" for the focal pattern 1 and a "2" for the second pattern 2.
- the fourth and fifth column carry the marks of the pattern 1 and 2 points, respectively (you can write the mark of the points in both columns).

6.5.2 Steps for data type 9 (DataType9.res)

Programita estimates for the univariate pattern of type 1 points the univariate mark correlation functions described under data type 6, and additionally the corresponding bivariate test statistics considering the marks of the two types of points. Because the null models randomize only within the component patterns, the univariate analysis is identical with the univariate analysis of the first component pattern.

To understand the bivariate mark correlation functions of data type 9 it is important to note that the bivariate mark correlation function for this data type estimates the mean value $c_t(r)$ of a test function $t(m_{i1}, m_{j2})$ taken over all pairs of points i1 of pattern 1 and j2 of pattern 2 distance r apart. The $c_t(r)$ is then normalized with the mean c_t taken over all pairs of points where the first point i1 is of pattern 1 and the second point j2 of pattern 2.

The bivariate methods are explained in detail in section 3.1.7.5 “Bivariate Pattern with One Quantitative Mark (Data Type 9)” in Wiegand and Moloney (2014). You can select the following “uni” and bivariate mark correlation functions:



Select one test function close
○ mark-correlation function ?
● r-mark correlation function m1 ?
○ r-mark correlation function m2 ?
○ mark variogram ?
○ Morans I mark statistic ?
○ Schlathers I ?
○ Density correlation ok
□ property ☑ cum □ surv

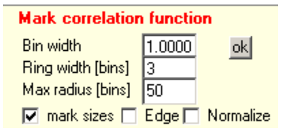
- the mark correlation function $k_{\mathrm{m1m2}}(r)$
- the r-mark correlation function $k_{\mathrm{m1}\,.}\,(r)$
- the r-mark correlation function $k_{.\,\mathrm{m2}}(r)$
- the mark variogram $\gamma_{\mathrm{m1m2}}(r)$
- a Moran’s I type mark statistics $I_{\mathrm{m1m2}}(r)$
- Schlather’s correlation function $I_{\mathrm{m1m2}}(r)$
- the density correlation functions $C_{m,g}(r)$ or $C_{m,K}(r)$
- the function $DD_2(r)$ that applies if the mark of the first pattern is binary (check “surv”).

Note that the two correlation coefficients $I_{\mathrm{m1m2}}(r)$ may show absolute values larger than one if the number of point pairs at this distance is very low (say < 10). To avoid this increase the ring width. Because the null models randomize within patterns, only the bivariate mark correlation functions are of interest here. For example, if the mark is the size of a tree

- the bivariate r-mark correlation function $k_{.\,\mathrm{m2}}(r)$ estimates the mean size of a tree of species 2 that has a focal tree of species 1 at distance r. This is the mark correlation analogue to trivariate random labeling because it investigates the impact of the presence of focal species 1 on the marking of trees of species 2 that are located at distance r of a focal tree. Here the null model must randomize the marks of pattern 2.
- the bivariate r-mark correlation function $k_{\mathrm{m1}\,.}\,(r)$ estimates the mean size of a tree of the focal species 1 that has a tree of species 2 at distance r. This analogue to trivariate random labeling allows you to use the density correlation function $C_{\mathrm{m1K2}}(r)$ to find out if the neighborhood density of points of the second pattern influences the marks of the focal pattern. Here the null model must randomize the marks of pattern 1.
- trivariate random labeling and DD_2 function: in this case the mark m_1 of pattern 1 is 1 (surviving) and 0 (dead). Enable the checkbox “surv”: ☑ surv
- the bivariate mark variogram $\gamma_{\mathrm{m1m2}}(r)$ estimates the mean squared size difference between a focal tree of species 1 and a tree of species 2 located at distance r. If the mark is size, this allows for example for an assessment of whether large focal tree of species 1 are surrounded by smaller trees the second species.

- the bivariate correlation function $I_{\mathrm{m1m2}}(r)$ estimates the correlation between the sizes of the trees of pattern 1 and that of trees of pattern 2 that are separated by distance r. The $I_{\mathrm{m1m2}}(r)$ therefore investigates the impact of the mark of the focal species 1 on the marking of nearby trees of species 2.
- the bivariate cumulative density correlation function $C_{\mathrm{m1K2}}(r)$ estimates the correlation between the mark of points i of pattern 1 and the density of the points of pattern 2 within distance r of points i. ☑ cum
- the bivariate non-cumulative density correlation function $C_{\mathrm{m1g2}}(r)$ estimates the correlation between the mark of points i of pattern 1 and the neighborhood density of the points of pattern 2 at distance r of points i. □ cum

The default for mark correlation functions are the normalized functions, i.e., $k_t(r) = c_t(r)/c_t$ where c_t is the normalization constant for a given test function t. However, you can also use the non-normalized mark correlation functions $c_t(r)$ when disabling the checkbox "**Normalize**".

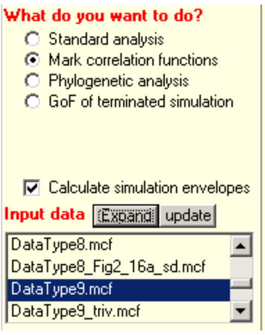

Mark correlation function
Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50
☑ mark sizes □ Edge □ Normalize

Independent marking for a bivariate pattern with one quantitative mark.

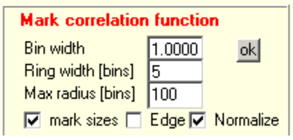
The mark correlation mode for a pattern with one qualitative and one quantitative mark can be accessed with the following sequence of actions:

1. Select "**Mark correlation functions**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file "DataType9.mcf".

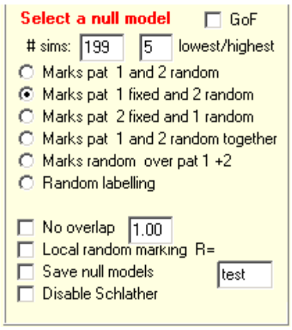
   This data file was generated by generating 600 random points (i.e., CSR), and then randomly selecting 200 points to be of the focal pattern 1. The mark m_1 of pattern 1 was then determined as $m_1 = 2 - \lambda_1 K_{11}(r = 25)/2.51$ and ranges between 0.01 and 2. Thus, the mark of pattern 1 was smaller if the point had more neighbors within 25m. The mark m_2 of pattern 2 was then determined as $m_2 = 2 - \lambda_1 K_{21}(r = 20)/2.51$ and ranges between 0.01 and 2. Thus, the mark of pattern 2 was smaller if the point of pattern 2 had more neighbors of species 1 within 20m.


What do you want to do?
○ Standard analysis
● Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation
☑ Calculate simulation envelopes
Input data Expand update
DataType8.mcf
DataType8_Fig2_16a_sd.mcf
DataType9.mcf
DataType9_triv.mcf

3. Click "**List with coordinates, no grid**" in MCFunction
4. Provide in the window mark correlation functions the bin width in data units (1), an appropriate ring width (5), and a maximal distance r of the analysis (100).
5. If you use a ring width of 1 unit, you can later use the function "**Combine replicates**" to load the results of the analysis, to change the ring width, and to use the corresponding cumulative summary function (see "View results of mark correlation analysis").
6. Disable "**Normalize**" if you want to use the non-normalized mark correlation functions. The default is "Normalize"


Mark correlation function
Bin width 1.0000 ok
Ring width [bins] 5
Max radius [bins] 100
☑ mark sizes □ Edge ☑ Normalize

7. Check "**Edge**" if you want to use the Ripley edge correction. Default is no edge correction. Note that edge correction is not required for mark correlation functions.
8. Check "**Calculate simulation envelopes**" in window What do you want to do? The subwindow "Select a null model" appears.
9. Select an appropriate null model, the number of simulations of the null model (199), and the rule for the simulation envelopes (5' lowest and highest). Note that not all null models are appropriate for all mark correlation function data type and that they must be selected with care.
   Select the null model "**Marks pat 1 fixed and 2 random**" that shuffles the quantitative mark randomly over the points of pattern 2 but holds the marks of pattern 1 unchanged.



Select a null model □ GoF

sims: 199 5 lowest/highest

○ Marks pat 1 and 2 random

● Marks pat 1 fixed and 2 random

○ Marks pat 2 fixed and 1 random

○ Marks pat 1 and 2 random together

○ Marks random over pat 1 +2

○ Random labelling

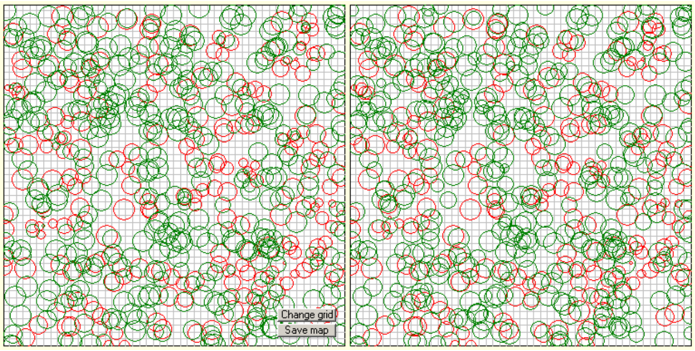
□ No overlap 1.00

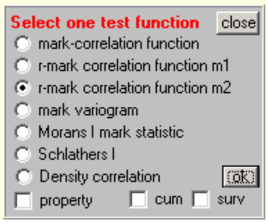
□ Local random marking R=

□ Save null models test

□ Disable Schlather

10. Press button "**Calculate Index**" and *Programita* shows the observed and simulated patterns. The red circles represent the mark of pattern 1 and the green circles the mark of pattern 2, and the area of the disk is proportional to the mark. Note that the null model changes only the size of pattern 2 (i.e., the green circles) but not that of pattern 1 (red circles):





Select one test function close

○ mark-correlation function

○ r-mark correlation function m1

● r-mark correlation function m2

○ mark variogram

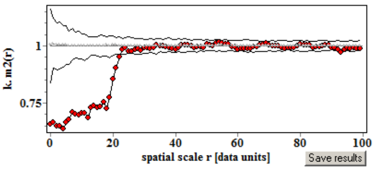
○ Morans I mark statistic

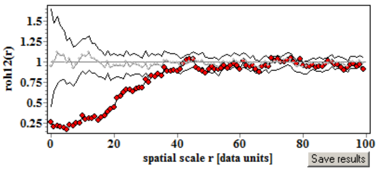
○ Schlathers I

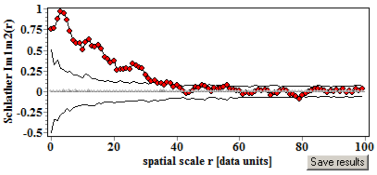
○ Density correlation ok

□ property □ cum □ surv

11. Use the radio buttons of the window Select one test function to select a mark correlation function and click the small "**ok**" button to get the result graphic:
12. Of special interest here is the *r*-mark correlation function $k_{.\,m2}(r)$ that estimates the mean size of a point of pattern 2 at distance *r* of a point of pattern 1. As expected, it reveals that the mark of pattern 2 is smaller than expected if the points is within 20m of a point of pattern 1:

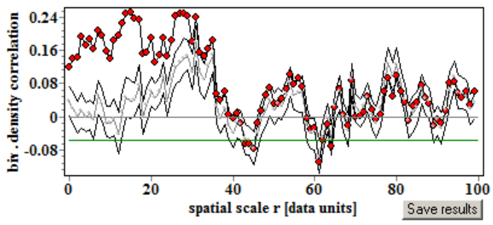


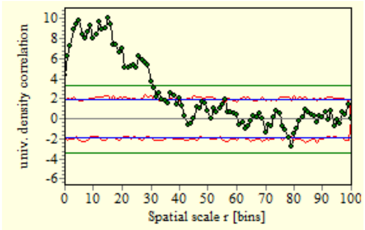




13. The mark variogram shows that the marks of the two patterns are more similar (i.e., smaller) if they are close together. This is because both, the marks of the points of pattern 1 and of pattern 2 were smaller if more points of pattern 1 were nearby. As a consequence of this, the marks of nearby points of the two patterns are positively correlated (Schlather's I).

14. If we enable the checkbox “property” but not “cum”, ☑ property □ cum □, the non-cumulative density correlation function correlates the size of the focal tree *i* of pattern 1 with the sum of the sizes of trees of pattern 2 in the ring with radius *r* and width *r* at distance r of point i. The resulting density correlation function looks somewhat unusual, but the student transformation of the “GoF” analysis shows that there is a strong positive correlation up to 30m between the size of the focal tree i of pattern 1 and the sum of the sizes of trees of pattern 2 (which is larger if the tree had more neighbors of species 1 within distance 20m) in the rings around the focal trees i of pattern 1:



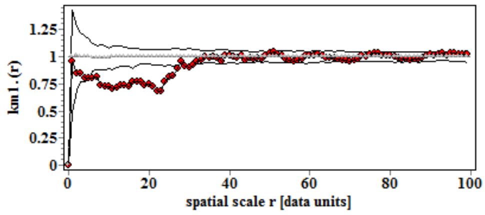


The correlation emerges because of the construction of the marks where the mark of pattern 1 was smaller if the point had more species 1 neighbors within 25m and the mark of pattern 2 if it had more neighbors of species 1 within 20m.

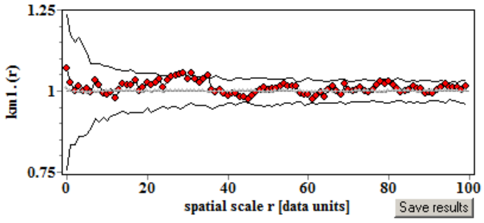
Randomize marks of pattern 1

15. Using the alternative null model for pattern DataType9.mcf where only the mark of pattern 1 is randomized (“**Marks pat2 fixed and 1 random**”) shows that points of pattern 1 which are located within distance 25m of another point of pattern 1 are smaller than expected, but that the mark of pattern 1 is not influenced by presence of a point of pattern 2:

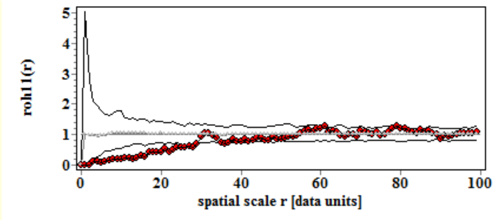
univariate

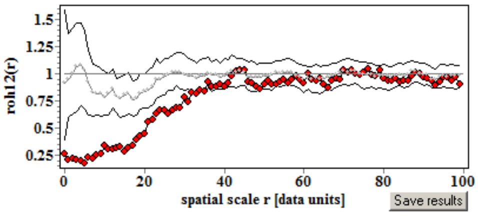


bivariate

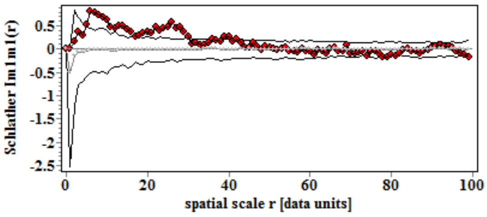


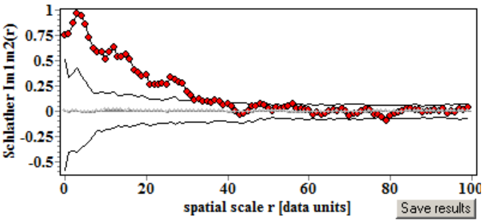
However, because both, the marks of the points of pattern 1 and that of pattern 2 depended on the number of neighbors of pattern 1, the mark variogram still depicts a significant bivariate effect:



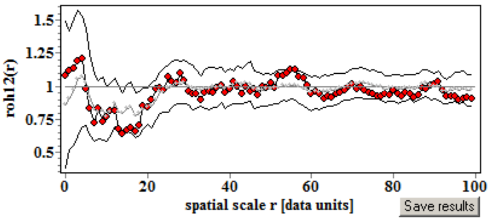


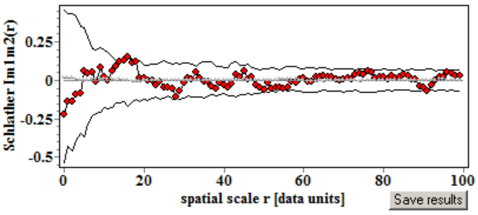
The same is true for Schlather’s I:





16. However, if we use the data set where the marks of pattern 1 were randomized (DataType9R1.mcf) and use null model **Marks pat2 fixed and 1 random** the latter two effects disappear because the spatial correlation between the marks of pattern 1 and pattern 2 was removed:





6.5.3 Trivariate random labeling with data type 9 (DataType9_triv.res)

The non-normalized versions of the *r*-mark correlation functions $k_{.\mathrm{m}2}(r)$ and $k_{\mathrm{m}1,.}(r)$ can be used together with the null models **Marks pat 1 fixed and 2 random** and **Marks pat 2 fixed and 1 random**, respectively, to conduct trivariate random labeling. That means the mark of interest which is carried by pattern 2 or pattern 1, respectively, is a binary mark such as surviving (value 1) and dead (value 0). Trivariate random labeling then investigates the impact of a second pattern on the marking of the pattern of interest. For example, using trivariate random labeling you can ask: what is the impact of proximity of large conspecific trees on survival of saplings? Is there (negative) density dependence operating where the survival of saplings depends on the neighborhood density of large trees?

The two non-normalized mark correlation functions are suitable for trivariate random labeling because they can estimate the mean value of the binary mark over all pairs of pattern 1 - pattern 2 points which are distance *r* apart. If the mark represents surviving (1) vs. dead (0) the r-mark correlation functions estimate the proportion of surviving trees among all pairs of trees that are located at distance *r* of a tree of the other pattern. Thus, the non-normalized r-mark correlation function $c_{.\mathrm{m}2}(r)$ estimates the proportion of surviving trees of pattern 2 among all pairs of pattern 1 - pattern 2 points which are distance *r* apart. Conversely, the non-normalized r-mark correlation function $c_{\mathrm{m}1.}(r)$ estimates the proportion of surviving trees of pattern 1 among all pairs of pattern 1 - pattern 2 points which are distance *r* apart. If the other pattern (e.g., large trees) has a negative impact on survival of saplings, the $c_{\mathrm{m}1.}(r)$ will be smaller than the mean mark of pattern 1 (i.e., the survival rate).

If the qualitative mark is carried by pattern 1, the $c_{\mathrm{m}1.}(r)$ is used and we can additionally estimate the two summary functions $DD_1(r)$ and $DD_2(r)$ that directly capture the effect of the density of pattern 1 (or pattern 2) neighbors on mortality of the focal pattern 1. Remember that the cumulative function $DD_1(r)$ and $DD_2(r)$ are the ratios:

$$DD_1(r) = \lambda_1 K_{\mathrm{d}1,1}(r) / \lambda_1 K_{\mathrm{s}1,1}(r)$$

$$DD_2(r) = \lambda_2 K_{\mathrm{d}1,2}(r) / \lambda_2 K_{\mathrm{s}1,2}(r)$$

where $\lambda_1 K_{\mathrm{d}1,1}(r)$ is the mean number of points of pattern 1 around dead individuals (subscript d) of pattern 1, $\lambda_2 K_{\mathrm{d}1,2}(r)$ is the mean number of points of pattern 2 around dead individuals of pattern 1, and the analogous quantities with subscript “s” instead of “d” are those around surviving individuals of pattern 1. The non-normalized bivariate *r*-mark correlation function $c_{\mathrm{m}1,.}(r)$, which yields the proportion of surviving trees of pattern 1 in all pairs of type 1 - type 2 points separated by distance *r*, is closely related with $DD_2(r)$:

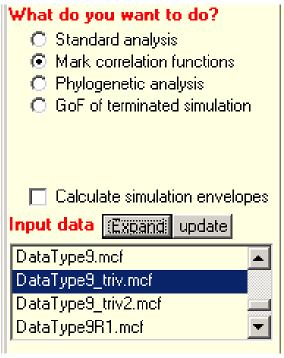
$$c_{\mathrm{m}1,.}(r) = \mathrm{s}/[DD_2(r)(1-s)+s]$$

where the mean mark of pattern 1 yields the survival rate *s*. Thus, departures of the bivariate $c_{\mathrm{m}1,.}(r)$ from the expected survival rate *s* occur only because of density dependence in mortality with respect to pattern 2.

Trivariate random labeling, Example DataType9_triv.res

We use in the first example the non-normalized *r*-mark correlation function $c_{.m2}(r)$ that estimates the mean mark of pattern 2 (e.g., survival of saplings) around points of pattern 1 (e.g., large trees). We use the data set DataType9.mcf, but make the marks of pattern 2 binary marks by defining them as 0 if the mark of DataType9.mcf was below 1.3 and 1 otherwise. Thus, trees with more neighbors of type 1 are more likely to die.

1. Select “**Mark correlation functions**” in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file “DataType9_triv.mcf”.
3. Click “**List with coordinates, no grid**” in MCFunction
4. Provide in the window mark correlation functions the bin width in data units (1), an appropriate ring width (5), and a maximal distance *r* of the analysis (50).
5. Check “**Calculate simulation envelopes**” in window What do you want to do? The subwindow “Select a null model” appears.
6. Select the null model “**Marks pat 1 fixed and 2 random**” that shuffles the mark of pattern 2 randomly over the points of pattern 2, the number of simulations of the null model (199), and the rule for the simulation envelopes (5’ lowest and highest).
7. De-select “**Normalize**” to obtain a quantity with the interpretation of a survival rate.
8. Press button “**Calculate Index**” and *Programita* simulates the null model.
9. As expected, the *r*-mark correlation function $k_{.m2}(r)$ shows that only 36% of all points of pattern 2 that are closer than 20m from a point of pattern 1 have mark 1 (surviving) as opposed by an average of 75% among all pattern 2 points:

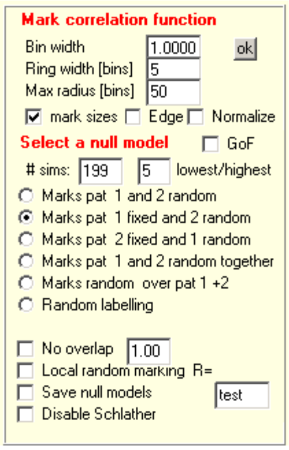


What do you want to do?
○ Standard analysis
● Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation

□ Calculate simulation envelopes

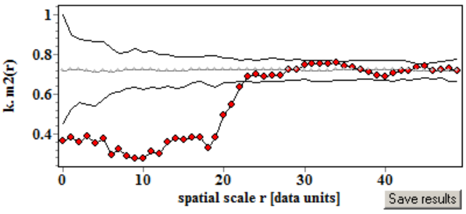
Input data

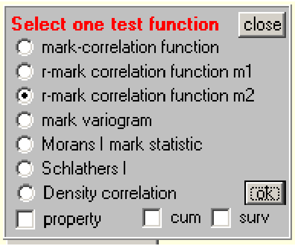
DataType9.mcf
DataType9_triv.mcf
DataType9_triv2.mcf
DataType9R1.mcf



Mark correlation function
Bin width 1.0000
Ring width [bins] 5
Max radius [bins] 50
☑ mark sizes □ Edge □ Normalize
Select a null model □ GoF
# sims: 199 5 lowest/highest
○ Marks pat 1 and 2 random
● Marks pat 1 fixed and 2 random
○ Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
○ Random labelling

□ No overlap 1.00
□ Local random marking R=
□ Save null models test
□ Disable Schlather



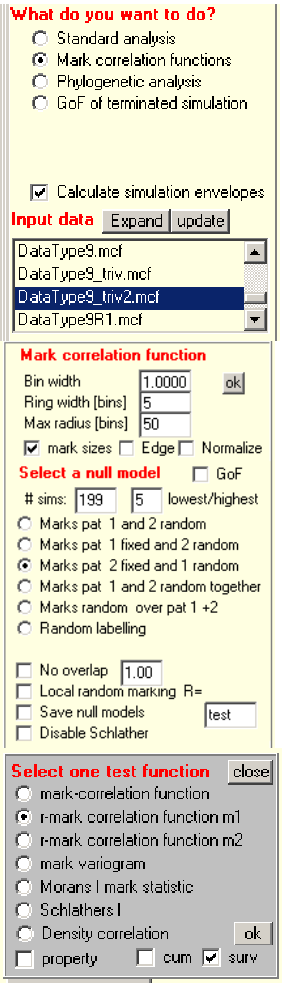


Select one test function
○ mark-correlation function
○ r-mark correlation function m1
● r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation
□ property □ cum □ surv

Trivariate random labeling, Example DataType9_triv2.res

To apply the density correlation function, the pattern with the binary mark must be the focal pattern 1 and the pattern that potentially influenced the binary mark must be pattern 2. We therefore use for this the same data set as before, but exchange patterns 1 and 2 (i.e., file DataType9_triv2.res). In this case the null model "**Marks pat 2 fixed and 1 random**" must be used.

1. Select "**Mark correlation functions**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. In the example it is file "DataType9_triv2.mcf".
3. Click "**List with coordinates, no grid**" in MCFunction
4. Provide in the window mark correlation functions the bin width in data units (1), an appropriate ring width (5), and a maximal distance *r* of the analysis (50).
5. Click "**Calculate Index**".
6. Check "**Calculate simulation envelopes**" in window What do you want to do? The subwindow "Select a null model" appears.
7. Select the null model "**Marks pat 2 fixed and 1 random**" that shuffles the mark of pattern 1 randomly over the points of pattern 1, the number of simulations of the null model (199), and the rule for the simulation envelopes (5' lowest and highest).
8. Check in Select one test function the check box ☑ surv.
9. De-select "**Normalize**" to obtain a quantity with the interpretation of a survival rate.
10. Press button "**Calculate Index**" and *Programita* simulates the null model.
11. As expected, the *r*-mark correlation function $k_{\mathrm{m1},\cdot}(r)$ shows that only 36% of all points of pattern 2 that are closer than 20m from a point of pattern 1 have mark 1 (surviving) as opposed by an average of 75% among all pattern 2 points (note that this result is identical to that above in example *DataType9_triv.res*):



What do you want to do?

○ Standard analysis
● Mark correlation functions
○ Phylogenetic analysis
○ GoF of terminated simulation

☑ Calculate simulation envelopes

Input data Expand update

DataType9.mcf
DataType9_triv.mcf
DataType9_triv2.mcf
DataType9R1.mcf

Mark correlation function

Bin width 1.0000 ok
Ring width [bins] 5
Max radius [bins] 50
☑ mark sizes □ Edge □ Normalize

Select a null model □ GoF

sims: 199 5 lowest/highest

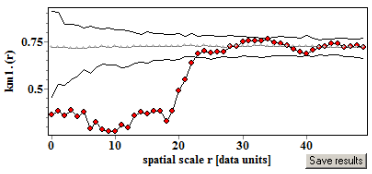
○ Marks pat 1 and 2 random
○ Marks pat 1 fixed and 2 random
● Marks pat 2 fixed and 1 random
○ Marks pat 1 and 2 random together
○ Marks random over pat 1 +2
○ Random labelling

□ No overlap 1.00
□ Local random marking R=
□ Save null models test
□ Disable Schlather

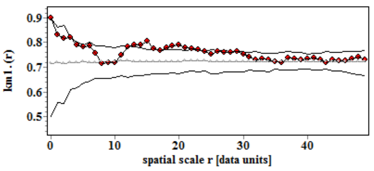
Select one test function close

○ mark-correlation function
● r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation ok
□ property □ cum ☑ surv

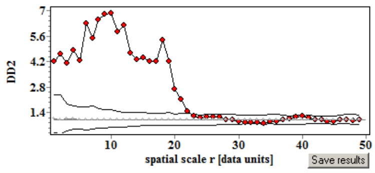
$c_{\mathrm{m1},\cdot}(r)$



non-cumulative $DD_1(r)$



non-cumulative $DD_2(r)$



however now we can estimate the ratio of the mean number of neighbors of pattern 2 around dead type 1 points relative to that of surviving type 1 points [i.e., the non-cumulative function $DD_2(r)$] which shows that the density of type 2 points around dead points of type 1 is on average 4-6 times as high as around surviving points. The non-cumulative $DD_2(r)$ also shows nicely that a density effect occurs only up to distances of 20m. As expected by the construction of the pattern, the univariate non-cumulative function $DD_1(r)$ shows that there is little to no effect of conspecific neighborhood density in the marking surviving vs. dead.

6.6 Combine replicates for mark correlation functions

In some cases you may have maps of several replicate plots of a larger point pattern under identical conditions. In this case the resulting test statistics of the individual replicate plots can be combined into average test statistics (Diggle 2003: page 123; Illian et al. 2008: page 263; Wiegand and Moloney 2014: section 3.2). This is of particular interest if the number of points in each replicate plot is relatively low. In this case the simulation envelopes of individual analyses would become wide, but combining the data of several replicate plots into average test statistics increases the sample size and thus narrows the simulation envelopes. Section 3.2.1 of Wiegand and Moloney (2014) provides details on the aggregation formulas for different summary functions and section 3.2.2 several examples.

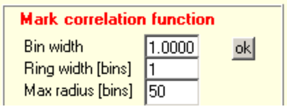
Combine replicates works for mark correlation functions in the same way as for the standard mode. However, the *.rep files are automatically created if you save the results of a mark correlation analysis or if you run a series of mark correlation analyses. In the first case they follow the convention “mcf_name.rep” and in the second case they follow the convention “s_mcf_name.rep” where the “s_” indicates series. The same applies for phylogenetic analysis, here the *.rep files follow the convention “mcf_name_phy.rep”.

Lines 8 and 9 of the *.res results file of mark correlation functions contain also the values of the normalization constants c_t for the different mark correlation functions which are needed in the aggregation formula for normalized mark correlation functions. For example, line 8 shows for a univariate analysis the following information:

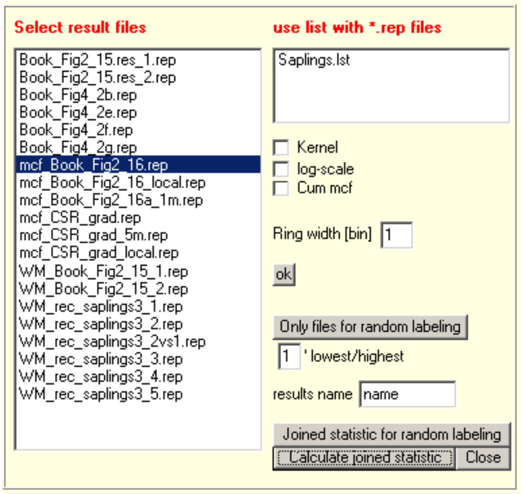
```
number points of pattern 1 =    103
mean mark p1=     77.2718
variance marks p1=  3910.3144
mean mark p1+p2=    77.2718
ct:  3910.3144   5970.9380     77.2718     77.2718   3910.3144   3910.3144
```

where “p1” refers to pattern 1 and the “ct” are the normalization constants of the six mark correlation functions.

Additionally to the standard mode, you can also change *a posteriori* the ring width, use the cumulative mark correlation function, or make the x-axis logarithmic. However, to manipulate the ring width or to make use the cumulative mark correlation function you need to run the analysis with a ring width of 1 to yield non-overlapping concentric rings.



Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50


Select result files
use list with *.rep files
ok
Only files for random labeling
Joined statistic for random labeling
Calculate joined statistic
Close

Book_Fig2_15.res_1.rep
Book_Fig2_15.res_2.rep
Book_Fig4_2b.rep
Book_Fig4_2e.rep
Book_Fig4_2f.rep
Book_Fig4_2g.rep
mcf_Book_Fig2_16.rep
mcf_Book_Fig2_16_local.rep
mcf_Book_Fig2_16a_1m.rep
mcf_CSR_grad.rep
mcf_CSR_grad_5m.rep
mcf_CSR_grad_local.rep
WM_Book_Fig2_15_1.rep
WM_Book_Fig2_15_2.rep
WM_rec_saplings3_1.rep
WM_rec_saplings3_2.rep
WM_rec_saplings3_2vs1.rep
WM_rec_saplings3_3.rep
WM_rec_saplings3_4.rep
WM_rec_saplings3_5.rep

Saplings.lst

□ Kernel
□ log-scale
□ Cum mcf

Ring width [bin] 1

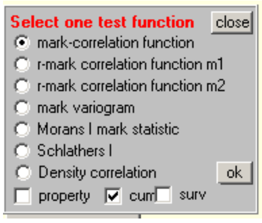
1 ' lowest/highest

results name name

Combine replicates for mark correlation functions

After each mark correlation analysis a temporary file MCF_test.dat is created which is then renamed into mcf_name.rep after saving results. The example below shows the first few lines of the univariate part of a mcf_name.rep output file:

```
simnr     r    MCF11_t0 MCF11_t1     MCF11_t2    MCF11_t3    MCF11_t4    MCF11_t5   Zaehler11
    0     0    1.00222  613.42544    0.99117     0.37239     0.37239     0.00155    964.00
    0     1    1.00321  613.76804    0.99173     1.25462     1.25462     0.00522    1298.00
    0     2    1.00255  612.62473    0.98988     2.45553     2.45553     0.01021    1664.00
```

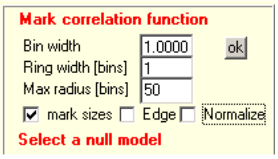


Select one test function close
● mark-correlation function
○ r-mark correlation function m1
○ r-mark correlation function m2
○ mark variogram
○ Morans I mark statistic
○ Schlathers I
○ Density correlation ok
□ property ☑ cum □ surv

The columns of the file contain the following information:

- simnr: number of simulation of the null model where 0 are the observed data and 1, 2, are the simulations of the null model.
- r: the distance bin

- MCF11_t0, MCF11_t1, …, MCF11_t5: the values of the six univariate mark correlation function where t1 refers to the “mark-correlation function”, t2 - t5 follow in descending order as shown in the selection window, and Schlather’s I is t0
- Zaehler11: the number of pairs at distance *r* (the denominator of the estimator equation (3.84) in Wiegand and Moloney (2014).
- MCF12_t0, MCF121_t1, …, MCF12_t5: the values of the six bivariate mark correlation functions.
- Zaehler12: the number of bivariate pairs at distance *r*.
- The density correlation functions CorDens11, CorDens11 and their normalization constants SDK11, SDK12, λ1K11, λ2K12



Mark correlation function
Bin width 1.0000 ok
Ring width [bins] 1
Max radius [bins] 50
☑ mark sizes □ Edge □ Normalize
Select a null model

Based on the information in this file, *Programita* can re-estimate the numerator and the denominator of equation (3.84) and apply the aggregation formula (3.107) in Wiegand and Moloney (2014).
If you use the default of normalized mark correlations, the information in lines 8 and 9 of the *.res file on the normalization constants c_t is used to “de-normalize” the mark correlation functions of the individual replicates by multiplying with c_t, and then the aggregation formula (3.107) is applied. The resulting non-normalized mark correlation functions are then normalized with a combined normalization constant c_t^{agg} .

The normalization constant c_t of a given replicate is the average of the test function $t(m_i, m_j)$ of all pairs of points of the pattern:

$$\hat{c}_t = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n,\neq} t(m_i, m_j)$$

where m_i, m_j are the marks of the points i and j of the pattern, respectively, and n the total number of points of the pattern. Therefore, we estimate the combined normalization constant c_t^{agg} as

$$\hat{c}_t^{agg} = \frac{\sum_{m=1}^{M} c_t^m (n_m - 1) n_m}{\sum_{m=1}^{M} (n_m - 1) n_m}$$

where the superscript *m* refers to replicate *m*. If you use non-normalized mark correlation functions, the results will be the same but not be divided by $\hat{c}_t^{agg}$.

7 Multivariate analysis

This analysis mode uses multivariate data (i.e., **data type 3**; several types of points; species) and multivariate summary functions that allow you to analyze spatial structures in diversity based on a dissimilarity matrix $d(f, m)$ that describes some distance between species f and species m, for example functional or phylogenetic distance or simply con-specific [$d(f, m) = 0$] vs. heterospecific [$d(f, m) = 1$].

Because such analyses involve two types of distances (spatial distance between individuals and dissimilarities between species), two fundamentally different type of null or point process models are possible:

(1) the **dissimilarity matrix is randomized**, but the locations and the individuals and their species identity remain unchanged. The methods to randomize the dissimilarity matrix are those of standard phylogenetic analysis (e.g., Hardy 2008). This task involves detection of small-scale spatial correlations in the dissimilarities of neighbored individuals, independent of the overall functional or phylogenetic community structure represented by *MPD* (i.e., obtain the mean pairwise dissimilarity c_d between all heterospecific individuals in the observation window *W*). The phylogenetic mark correlation function $k_\mathrm{d}(r)$ is especially suitable for this task because it is normalized with *MPD* (eq. 4) (Shen et al. 2013). **This type of analysis is described here in chapter 7.**

(2) The **locations of the individuals are randomized**, but the dissimilarity matrix remains unchanged. Here the task is to randomize or maintain certain features of the individual species patterns to determine the relative importance of different mechanism and processes (such as habitat association or dispersal limitation) with respect to the emerging community level spatial diversity patterns. This task is complex because appropriate null communities (or null models) must be created that possibly involve separate randomizations for the individuals of each species. However, for randomization of individual locations, we can take advantage of abundant techniques of point process modeling. **This type of analysis is described in chapter 8.**

7.1 General framework for multivariate point pattern diversity metrics

Programita generalizes two basic diversity indices, species richness *S* and the Simpson index *D*. **Species richness *S*** gives the total number of species in a (fully mapped) observation window *W* and the **Simpson index *D*** the probability that two randomly selected individuals in *W* are heterospecific. The general framework considers generalization of *S* and *D* into two dimensions, considering spatial distance and phylogenetic/functional dissimilarity.

First, a **continuous measure δ_{ij}^{P} of ecological dissimilarity** between species *i* and *j* (e.g., Clarke & Warwick 1998) is introduced that can represent functional dissimilarity, phylogenetic dissimilarity or any other ecological dissimilarity.

Second, *S* and *D* is **made truly spatially explicit** by using spatial point pattern methods (Wiegand & Moloney 2014; Wiegand et al. 2017) that look at pairs of individuals that are a given distance *r* apart (for metrics of beta diversity) or that are located within a given distance *r* (for metrics of alpha diversity). By applying this framework, eight families of spatially explicit diversity metrics arise depending on (i) whether they present alpha or beta diversity, (ii) whether they are based on *S* or *D*, and (iii) whether they quantify diversity from the perspective of individual species or from the perspective of the entire community.

Additionally, within each family, different diversity metrics emerge depending on whether they consider (A) species diversity, (B) functional or phylogenetic diversity, or (C) functional or phylogenetic diversity relative to species diversity:

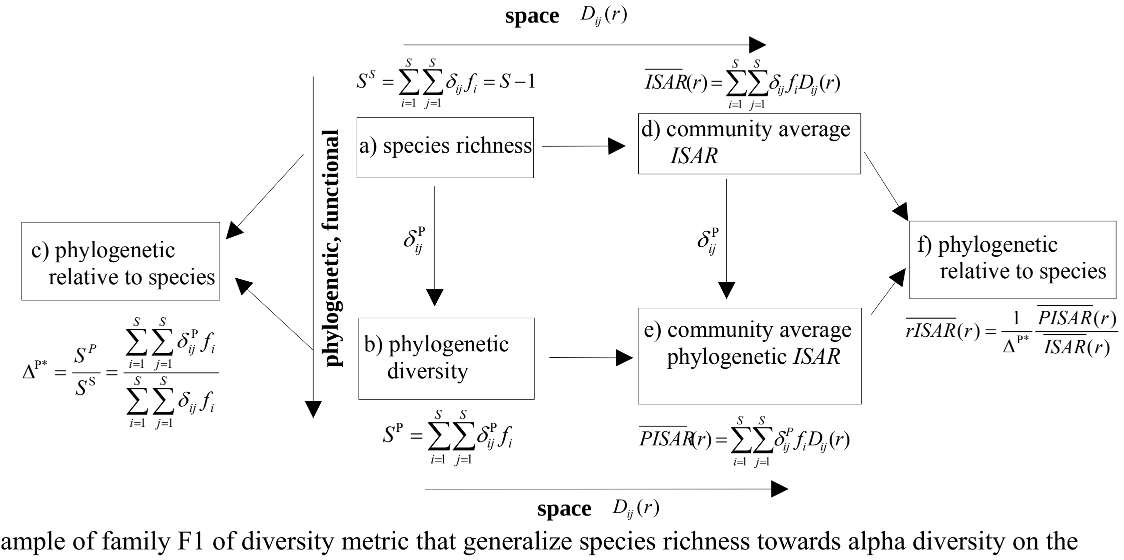
classifier				non-spatial metrics			spatial metrics			spatial
				A	B	C	A	B	C	condition
F1	α	S	community	S^S	S^P	$\Delta^{P*} = S^P/S^S$	$\overline{ISAR}(r)$	$\overline{PISAR}(r)$	$\overline{rISAR}(r)$	$D_{fs}(r)$
F2	α	S	focal species	S_f	S_f^P	$\Delta_f^P = S_f^P/S_f$	$ISAR_f(r)$	$PISAR_f(r)$	$rISAR_f(r)$	$D_{fs}(r)$
F3	α	D	community	D	D^P	$c_d = D^P/D$	$\alpha_S(r)$	$\alpha_{phy}(r)$	$K_d(r)$	$K_{ij}(r)/K(r)$
F4	α	D	focal species	D_f	D_f^P	$c_{fd} = D_f^P/D_f$	$\alpha_{f,S}(r)$	$\alpha_{f,phy}(r)$	$K_{f,d}(r)$	$K_{ij}(r)/K(r)$
F5*	β	S	community	S^S	S^P	$\Delta^{P*} = S^P/S^S$	$\overline{isar}(r)$	$\overline{pisar}(r)$	$\overline{risar}(r)$	$d_{fs}(r)$
F6*	β	S	focal species	S_f	S_f^P	$\Delta_f^P = S_f^P/S_f$	$isar_f(r)$	$pisar_f(r)$	$risar_f(r)$	$d_{fs}(r)$
F7	β	D	community	D	D^P	$c_d = D^P/D$	$\beta_S(r)$	$\beta_{phy}(r)$	$k_d(r)$	$g_{ij}(r)/g(r)$
F8	β	D	focal species	D_f	D_f^P	$c_{fd} = D_f^P/D_f$	$\beta_{f,S}(r)$	$\beta_{f,phy}(r)$	$k_{f,d}(r)$	$g_{ij}(r)/g(r)$

*Metric families F5 and F6 that are based on the non-cumulative probability density function $d_{ij}(r)$ of the distances to the nearest species j neighbor have not been used to date.

The non-spatial diversity metrics are made spatially explicit by **adding a spatial condition** that is expressed by means of point pattern summary functions such as

- the probability $D_{ij}(r)$ that an individual of species j is located within distance r of an individual of species i (families F1 and F2)
- the conditional probability $f_i f_j K_{ij}(r)/g(r)$ that of two randomly selected individuals within distance r the first belong to species i and the second to species j (F3 and F4)
- the conditional probability $f_i f_j g_{ij}(r)/g(r)$ that of two randomly selected individuals at distance r the first belong to species i and the second to species j (F7 and F8)

where f_i is the relative abundance of species i. Note that the spatial metrics are possibly normalized by the corresponding non-spatial metrics. If the dissimilarity matrix is randomized, normalization of the metrics is important.


space $D_{ij}(r)$
$S^S = \sum_{i=1}^{S}\sum_{j=1}^{S}\delta_{ij} f_i = S - 1$
$\overline{ISAR}(r) = \sum_{i=1}^{S}\sum_{j=1}^{S}\delta_{ij} f_i D_{ij}(r)$
a) species richness
d) community average ISAR
phylogenetic, functional
c) phylogenetic relative to species
δ_{ij}^P
δ_{ij}^P
f) phylogenetic relative to species
$\Delta^{P*} = \frac{S^P}{S^S} = \frac{\sum_{i=1}^{S}\sum_{j=1}^{S}\delta_{ij}^P f_i}{\sum_{i=1}^{S}\sum_{j=1}^{S}\delta_{ij} f_i}$
b) phylogenetic diversity
e) community average phylogenetic ISAR
$\overline{rISAR}(r) = \frac{1}{\Delta^{P*}} \frac{\overline{PISAR}(r)}{\overline{ISAR}(r)}$
$S^P = \sum_{i=1}^{S}\sum_{j=1}^{S}\delta_{ij}^P f_i$
$\overline{PISAR}(r) = \sum_{i=1}^{S}\sum_{j=1}^{S}\delta_{ij}^P f_i D_{ij}(r)$
space $D_{ij}(r)$

Example of family F1 of diversity metric that generalize species richness towards alpha diversity on the community level (see table above). The scheme shows the relationships among classical and spatially explicit metrics of species richness S where generalization towards spatially-explicit metrics is shown along the horizontal axis and generalization towards a measure of phylogenetic (or functional) diversity along the vertical axis

7.1.1 Metrics of species diversity, *ISAR* family

The individual species-area relationship $ISAR_f(r)$ (Wiegand et al. 2007) views the local species richness of the community from the viewpoint of an individual focal species *f* (family F2 in the table above). It is defined as the **expected species richness with radius *r* of the typical individual of the focal species *f***:

$$ISAR_f(r) = \sum_{j=1}^{S} D_{fj}(r)\delta_{fj}$$

where $D_{fj}(r)$ is the probability that the nearest species *j* neighbor of an individual of the focal species *f* is located within distance *r*, and δ_{fj} is zero for conspecifics (i.e., $f = j$) and one for heterospecifics (i.e., $f \neq j$). The $ISAR_f(r)$ is the point pattern analog of the species area relationships where the sample areas are circles of varying radius around the individuals of a given focal species *f*.

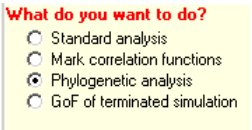
The individual species-area relationship can also be averaged over all species present in the community (family F1 in the table above) by summing up the $ISAR_i(r)$'s for all species *i* and weighting with their relative abundance f_i:

$$\overline{ISAR}(r) = \sum_{i=1}^{S} f_i ISAR_i(r) = \sum_{i=1}^{S} f_i \sum_{i=1}^{S} D_{ij}(r)\delta_{ij}$$

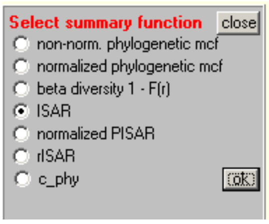
The community averaged individual species area relationship $\overline{ISAR}(r)$ **is the expected species richness within radius *r* of the typical individual of the community**.

The *ISAR* function has an intuitive **geometric interpretation** for a fixed neighborhood radius *r*. If we count for each location **x** the number of species within distance *r* we obtain a "landscape of local species richness" with valleys (low neighborhood species richness) and mountain ridges (high neighborhood species richness). The *ISAR* function shows then how the focal species *f* is embedded within this landscape: species located mostly in valleys are "repeller" species (surrounded by species poorer assemblages), whereas species in ridges are "accumulators" (typically surrounded by species richer assemblages).

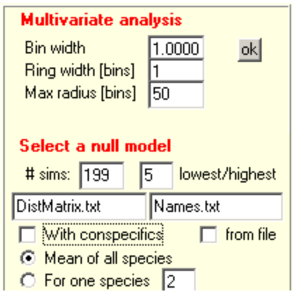
The *ISAR* functions can be accessed in *Programita* by selecting "Phylogenetic analysis", "Mean of all species" (for the community level ISAR) or "For one species" (for the ISAR) in the "Multivariate analysis" window and selecting "ISAR" in the "summary function" window. You can calculate it with and without considering conspecifics:



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

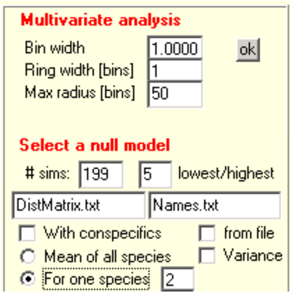


Select summary function
○ non-norm. phylogenetic mcf
○ normalized phylogenetic mcf
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50
Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2

Community level



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50
Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 2

Individual level, specify species (2 in the example)

Note that the ISAR function, which is measuring species richness, is insensitive to the randomization of the dissimilarity matrix (that leaves the locations and species identity of the individuals unchanged). However, the ISAR will become important in the null community approaches detailed in chapter 8.

7.1.2 Metrics of species diversity, Simpson alpha diversity family

The cumulative spatially-explicit Simpson index (Shimatani 2001) represents alpha diversity at the community level (family F3 in the table above) and is the **probability that two individuals within distance *r* are heterospecifics**. It can be estimated as:

$$\alpha_S(r)=\sum_{i=1}^{S} f_i \sum_{j=1}^{S} \delta_{ij} f_j \frac{K_{ij}(r)}{K(r)}$$

where $K_{ij}(r)$ is the partial *K* function of the species pair *i*-*j*, and $K(r)$ is that of all individuals within *W*. For more detail see section 3.15 in Wiegand and Moloney (2014).

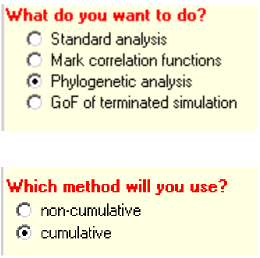
The cumulative spatially-explicit Simpson index at the focal species level (family F4 in the table above) **is the proportion of heterospecific neighbors within distance *r* of the typical individuals of the focal species *f*** and given by

$$\alpha_{f,S}(r)=\sum_{j=1}^{S} \delta_{fj} f_j \frac{K_{fj}(r)}{K_{fa}(r)} = 1-f_f \frac{K_{ff}(r)}{K_{fa}(r)}$$

where the $K_{fa}(r)$ is the partial *K* function of the focal species *f* to all neighbors (i.e., the subscript *a*).

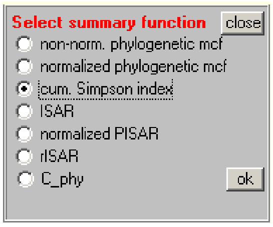
Note that the cumulative spatially-explicit Simpson index $\alpha_{f,S}(r)$ at the focal species level can also be expressed as one minus the proportion of conspecifics within distance *r* of the typical individuals of the target species. Thus, the metric $L_t(r) = 1–\alpha_{f,S}(r)$ is an **index of local dominance of the focal species *f*** (Wiegand et al. 2007, Wiegand & Moloney 2014) because it is the number of conspecific neighbors of the typical individual of the focal species within distance *r* [$\lambda_f K_{ff}(r)$] divided by its total number of neighbors within distance *r* [$\lambda K_{fa}(r)$; $f_\mathrm{f} = \lambda_f/\lambda$]. In other words, $L_t(r) = 1 – \alpha_{t,S}(r)$ yields the mean proportion of conspecific neighbors within a neighborhood of radius *r* around the typical individuals of the target species *t*.

The cumulative spatially-explicit Simpson index can be accessed in *Programita* by selecting “Phylogenetic analysis”, “cumulative” in the window “Which method will you use”, “With conspecifics” and “Mean of all species” (for the community level function) or “For one species” (for the individual level function) in the “Multivariate analysis” window and selecting “cum. Simpson index” in the “summary function” window:



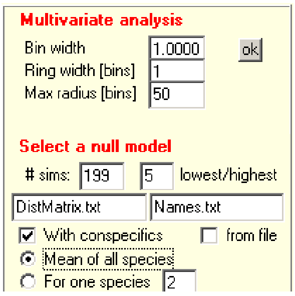
What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

Which method will you use?
○ non-cumulative
● cumulative



Select summary function

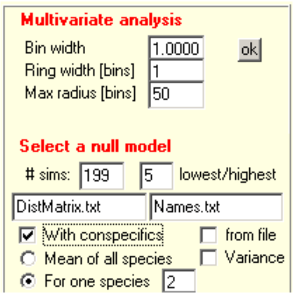
○ non-norm. phylogenetic mcf
○ normalized phylogenetic mcf
● cum. Simpson index
○ ISAR
○ normalized PISAR
○ rISAR
○ C_phy



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
☑ With conspecifics ☐ from file
● Mean of all species
○ For one species 2

Community level



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
☑ With conspecifics ☐ from file
○ Mean of all species ☐ Variance
● For one species 2

Individual level, specify species (2 in the example)

Note that the cumulative Simpson index is insensitive to the randomization of the dissimilarity matrix (that leaves the locations and species identity of the individuals unchanged). However, it will become important in the null community approaches detailed in chapter 8.

7.1.3 Metrics of species diversity, Simpson beta diversity family

The spatially-explicit Simpson index (Shimatani 2001) represents beta diversity at the community level (family F7 in the table above) and is the **probability that two individuals at distance *r* are heterospecifics**. It can be estimated as:

$$\beta_S(r) = \sum_{i=1}^{S} f_i \sum_{j=1}^{S} \delta_{ij} f_j \frac{g_{ij}(r)}{g(r)}$$

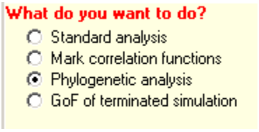
where $g_{ij}(r)$ is the partial pair correlation function of the species pair *i*-*j*, and $g(r)$ is that of all individuals within *W*. For more detail see section 3.15 in Wiegand and Moloney (2014). The **$\beta_S(r)$ is an index of spatial species turnover between locations** that indicates how species dissimilarity changes with distance *r* between locations of the local community. Note that $\beta_S(r)$ is directly related to the well established metric $F(r)$ of beta diversity used in Chave and Leigh (2002) and Condit et al. (2002) by $F(r) = 1 - \beta_S(r)$ where $F(r)$ describes the distance decay of species similarity.

The spatially-explicit Simpson index at the focal species level (family F8 in the table above) **is the proportion of heterospecific neighbors at distance *r* of the typical individuals of the focal species *f*** and given by

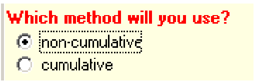
$$\beta_{f,S}(r) = \sum_{j=1}^{S} \delta_{fj} f_j \frac{g_{fj}(r)}{g_{fa}(r)} = 1 - f_f \frac{g_{ff}(r)}{g_{fa}(r)}$$

where the $g_{fa}(r)$ is the partial pair correlation function of the focal species *f* to all neighbors (i.e., the subscript *a*). Thus, **$\beta_{t,S}(r)$ captures local dominance of the target species *t* in a ring with radius *r* around the focal individuals**.

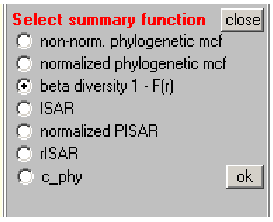
The spatially-explicit Simpson index can be accessed in *Programita* by selecting "Phylogenetic analysis", "cumulative" in the window "Which method will you use", "With conspecifics" and "Mean of all species" (for the community level function) or "For one species" (for the individual level function) in the "Multivariate analysis" window, and selecting "beta diversity 1 - F(r)" in the "summary function" window:



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

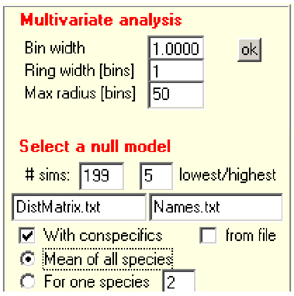


Which method will you use?
● non-cumulative
○ cumulative



Select summary function

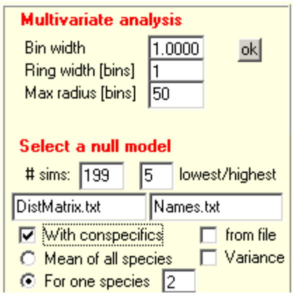
○ non-norm. phylogenetic mcf
○ normalized phylogenetic mcf
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2

Community level



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
☑ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 2

Individual level, specify species (2 in the example)

Note that the cumulative Simpson index is insensitive to the randomization of the dissimilarity matrix (that leaves the locations and species identity of the individuals unchanged). However, it will become important in the null community approaches detailed in chapter 8.

7.1.4 Metrics of phylogenetic diversity, *ISAR* family

The individual species-area relationship $ISAR_f(r)$ can be extended towards a measure of the local functional or phylogenetic diversity around a given target species (Wang et al. 2016). This can be done by replacing the binary dissimilarity measure δ_{ij} by a continuous measure δ_{ij}^{P} of pairwise species dissimilarity (Wiegand et al. 2017). We then obtain the **sum of all pairwise phylogenetic distances of the typical individual of the focal species *f* to all other species within distance *r***:

$$PISAR_f(r)=\sum_{j=1}^{S}\delta_{fj}^{P}D_{fj}(r)\,,\ \text{ can be normalized with } S_f^{\mathrm{P}}=\sum\nolimits_{j=1}^{S}\delta_{fj}^{\mathrm{P}}$$

where $D_{fj}(r)$ is the probability that the nearest species *j* neighbor of an individual of species *f* is located within distance *r*. A similar approach to extend the *ISAR* function, but based on phylogenetic diversity *PD* (Faith 1992), has been proposed by Yang et al. (2013).

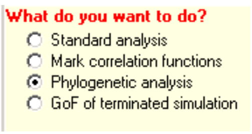
The phylogenetic individual species area relationship *PISAR* can also be averaged over all species present in the community (family F1 in the table above) by summing up the $PISAR_i(r)$'s for all species *i* and weighting with their relative abundance f_i:

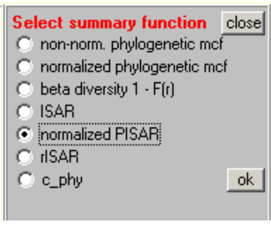
$$\overline{PISAR}(r)=\sum_{i=1}^{S}f_iPISAR_i(r)=\sum_{i=1}^{S}f_i\sum_{i=1}^{S}\delta_{ij}^{P}D_{ij}(r)\ \text{ can be normalized with } S^{\mathrm{P}}=\sum_{i=1}^{S}f_i\sum_{i=1}^{S}\delta_{ij}^{P}$$

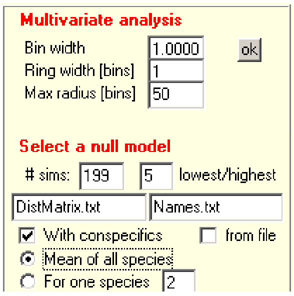
The community averaged phylogenetic species area relationship $\overline{PISAR}(r)$ therefore **is the sum of the pairwise dissimilarities of the typical individual of the community to all other species within distance *r***.

The *PISAR* function can also be reinterpreted to be able to analyze genetic variability within a given species. For a genotyped population the dissimilarity matrix then gives the genetic dissimilarity between individuals (not species) and the $PISAR_f(r)$ is the genetic diversity of individuals neighboring the focal individual *f* across increasingly large spatial distance *r*. Shao et al. (2018) called this function "**individual genetic diversity area relationship**" (IGDAR).

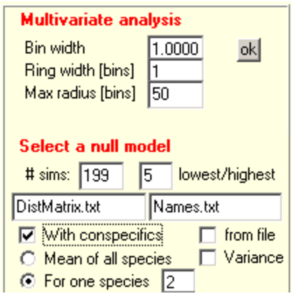
The *PISAR* functions can be accessed in *Programita* by selecting "Phylogenetic analysis", "Mean of all species" (for the community level PISAR) or "For one species" (for the PISAR) in the "Multivariate analysis" window and selecting "PISAR" in the "summary function" window. You can calculate it with and without considering conspecifics:


What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation


Select summary function [close]
○ non-norm. phylogenetic mcf
○ normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
● normalized PISAR
○ rISAR
○ c_phy [ok]


Multivariate analysis
Bin width 1.0000 [ok]
Ring width [bins] 1
Max radius [bins] 50
Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2

Community level


Multivariate analysis
Bin width 1.0000 [ok]
Ring width [bins] 1
Max radius [bins] 50
Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
☑ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 2

Individual level, specify species (2 in the example)

7.1.5 Metrics of phylogenetic diversity, α_S family

The phylogenetic extension $\alpha_{\text{phy}}(r)$ of the cumulative spatially-explicit Simpson index results from replacing the binary dissimilarity measure δ_{ij} by a continuous measure δ_{ij}^{P} of pairwise species dissimilarity. The $\alpha_{\text{phy}}(r)$ is the **expected dissimilarity of two randomly selected individuals in *W* that are located within distance *r***:

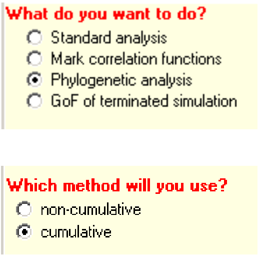
$$\alpha_{phy}(r) = \sum_{i=1}^{S} f_i \sum_{j=1}^{S} \delta_{ij}^{P} f_j \frac{K_{ij}(r)}{K(r)} \quad \text{can be normalized with} \quad D^{\text{P}} = \sum_{i=1}^{S}\sum_{j=1}^{S} \delta_{ij}^{P} f_i f_j$$

where $K_{ij}(r)$ is the partial K function of the species pair i-j, and $K(r)$ is that of all individuals within W. The $\alpha_{\text{phy}}(r)$ is a point pattern version of the within community version D^{P} of Rao’s quadratic entropy because it basically estimates the average of D^{P} for circular plots with radius r centered on all individuals of the community.

The cumulative phylogenetic Simpson index at the focal species level (family F4 in the table above) **is the mean dissimilarity of all individuals located within distance *r* of the typical individuals of the focal species *f*** and given by

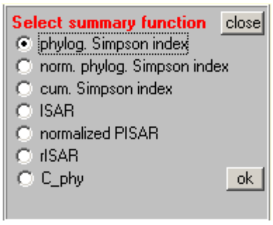
$$\alpha_{f,\text{S}}(r) = \sum_{j=1}^{S} \delta_{fj}^{P} f_j \frac{K_{fj}(r)}{K_{fa}(r)} \quad \text{can be normalized with} \quad D_f^{\text{P}} = \sum_{j=1}^{S} \delta_{fj}^{P} f_j$$

The cumulative spatially-explicit Simpson index can be accessed in *Programita* by selecting “Phylogenetic analysis”, “cumulative” in the window “Which method will you use”, “With conspecifics” and “Mean of all species” (for the community level function) or “With conspecifics” and “For one species” (for the individual level function) in the “Multivariate analysis” window and selecting “phylog. Simpson index” in the “summary function” window:



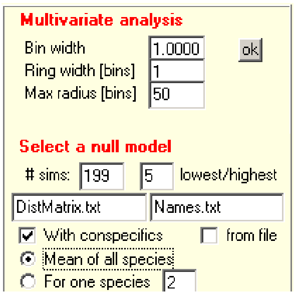
What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

Which method will you use?
○ non-cumulative
● cumulative



Select summary function

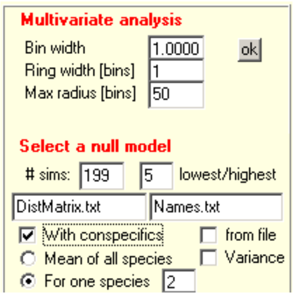
● phylog. Simpson index
○ norm. phylog. Simpson index
○ cum. Simpson index
○ ISAR
○ normalized PISAR
○ rISAR
○ C_phy



Multivariate analysis
Bin width: 1.0000
Ring width [bins]: 1
Max radius [bins]: 50

Select a null model
# sims: 199 | 5 | lowest/highest
DistMatrix.txt | Names.txt
☑ With conspecifics ☐ from file
● Mean of all species
○ For one species 2

Community level



Multivariate analysis
Bin width: 1.0000
Ring width [bins]: 1
Max radius [bins]: 50

Select a null model
# sims: 199 | 5 | lowest/highest
DistMatrix.txt | Names.txt
☑ With conspecifics ☐ from file
○ Mean of all species ☐ Variance
● For one species 2

Individual level, specify species (2 in the example)

7.1.6 Metrics of phylogenetic beta diversity, β_S family

The phylogenetic extension $\beta_{\mathrm{phy}}(r)$ of the spatially-explicit Simpson index results from replacing the binary dissimilarity measure δ_{ij} by a continuous measure δ_{ij}^{P} of pairwise species dissimilarity (Wang et al. 2015). The $\beta_{\mathrm{phy}}(r)$ is the **expected dissimilarity of two randomly selected individuals in *W* that are located at distance *r***:

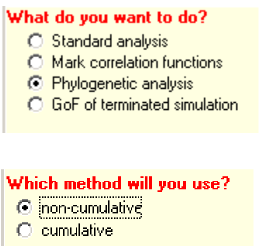
$$\beta_{phy}(r) = \sum_{i=1}^{S} f_i \sum_{j=1}^{S} \delta_{ij}^{P} f_j \frac{g_{ij}(r)}{g(r)} \text{ , can be normalized with } D^{\mathrm{P}} = \sum_{i=1}^{S} \sum_{j=1}^{S} \delta_{ij}^{P} f_i f_j$$

where $g_{ij}(r)$ is the partial pair correlation function of the species pair *i*-*j*, and $g(r)$ is that of all individuals within *W*. The **$\beta_S(r)$ is an index of spatial functional or phylogenetic turnover between locations** that indicates how dissimilarity changes with distance *r* between locations of the local community. The beta diversity metrics $\beta_S(r)$ and $\beta_{\mathrm{phy}}(r)$ are especially useful to test different (null) hypotheses regarding the mechanism underlying the placement of species (i.e., null communities; e.g., Wang et al. 2015).

The phylogenetic Simpson index at the focal species level (family F8 in the table above) **is the mean dissimilarity of all individuals located at distance *r* of the typical individuals of the focal species *f*** and given by

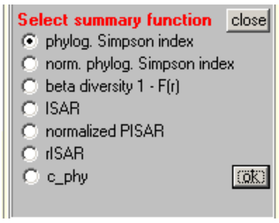
$$\beta_{f,S}(r) = \sum_{j=1}^{S} \delta_{fj}^{P} f_j \frac{g_{fj}(r)}{g_{fa}(r)} \text{ can be normalized with } D_f^{\mathrm{P}} = \sum_{j=1}^{S} \delta_{fj}^{P} f_j$$

The spatially-explicit Simpson index can be accessed in *Programita* by selecting "Phylogenetic analysis", "non-cumulative" in the window "Which method will you use", "With conspecifics" and "Mean of all species" (for the community level function) or "With conspecifics" and "For one species" (for the individual level function) in the "Multivariate analysis" window and selecting "phylog. Simpson index" in the "summary function" window



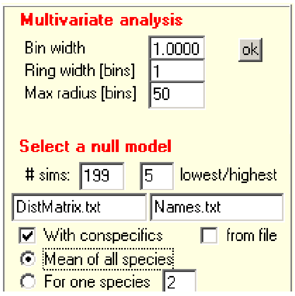
What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

Which method will you use?
● non-cumulative
○ cumulative



Select summary function

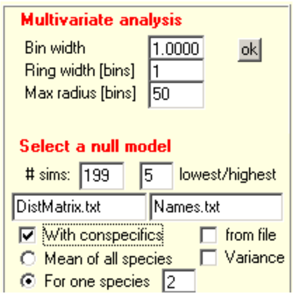
● phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
☑ With conspecifics ☐ from file
● Mean of all species
○ For one species 2

Community level



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
☑ With conspecifics ☐ from file
○ Mean of all species ☐ Variance
● For one species 2

Individual level, specify species (2 in the example)

7.1.7 Metrics of phylogenetic relative to species diversity, *ISAR* family

The *PISAR* function is often strongly determined by the underlying pattern of neighborhood species richness. This is unimportant if the dissimilarity matrix is randomized (because in this case the underlying *ISAR* function is unaffected by the null model), however, if the observed data are compared to null communities that randomize the locations of the individuals it will be difficult to tease apart the influence of species dissimilarities and species placement on the *PISAR*. An elegant way of factoring out the signal of the underlying *ISAR* in the *PISAR* is to use the *rISAR* function which is the *PISAR* divided by the *ISAR* (Wang et al. 2016; Wiegand et al. 2017):

$$rISAR_f(r) = \frac{PISAR_f(r)}{ISAR_f(r)}\text{, can be normalized with }\Delta_f^{\mathrm{P}} = \tfrac{1}{S-1}\sum_{j=1}^{S}\delta_{fj}^{\mathrm{P}}$$

The *rISAR* is the **expected pairwise dissimilarity between the typical individual of the focal species and all other species within distance *r*.** The normalization constant $\Delta^{\mathrm{P}}{}_{\mathrm{f}}$ is the mean pairwise dissimilarity of the focal species *f* to all other species in the observation window *W*. It is analogous to the community-level index Δ^{P} in Hardy and Senterre (2007) that measures overall phylogenetic distinctness based on species incidence within a given community. $\Delta^{\mathrm{P}}{}_{\mathrm{f}}$ is a measure of the distinctness of the focal species to all other species in *W*.

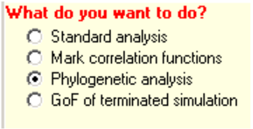
The community level version of the *rISAR* is given by

$$\overline{rISAR}(r) = \frac{\overline{PISAR}(r)}{\overline{ISAR}(r)}\text{, can be normalized with }\Delta^{\mathrm{P}*} = \sum\nolimits_{i,j=1}^{S}\delta_{ij}^{\mathrm{P}} f_i \Big/ \sum\nolimits_{i,j=1}^{S}\delta_{ij} f_i$$

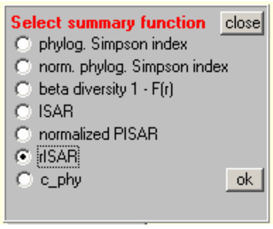
The community level *rISAR* is the **expected pairwise dissimilarity between the typical individual of the community and all other species within distance *r*.**

The normalization constant $\Delta^{\mathrm{P}*}$ views the community from the typical individual of the community and is therefore a measure of the distinctness of the typical individual of the community with respect to all other species in *W*.

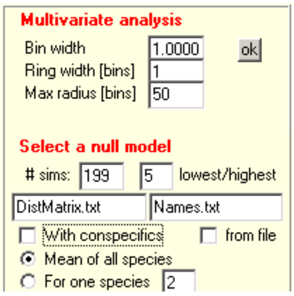
The *rISAR* can be accessed in *Programita* by selecting “Phylogenetic analysis”, “Mean of all species” (for the community level rISAR) or “For one species” (for the *rISAR*) in the “Multivariate analysis” window and selecting “rISAR” in the “summary function” window:



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

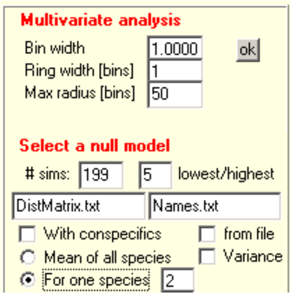


Select summary function [close]
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
● rISAR
○ c_phy
[ok]



Multivariate analysis
Bin width 1.0000 [ok]
Ring width [bins] 1
Max radius [bins] 50
Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2

Community level



Multivariate analysis
Bin width 1.0000 [ok]
Ring width [bins] 1
Max radius [bins] 50
Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 2

Individual level, specify species (2 in the example)

The species-level *rISAR* function has also an **intuitive graphical interpretation**. For each location **x** in the observation window we can determine the mean dissimilarity $\Delta_t^{\mathrm{P}}(\mathbf{x}, r)$ of the focal species *f* to all other species within distance *r*. The resulting map shows, from the viewpoint of the focal species *f*, areas with more similar assemblages compared to a spatially random community (i.e., $\Delta_t^{\mathrm{P}}(\mathbf{x}, r) < \Delta_t^{\mathrm{P}}$) and areas with more dissimilar assemblages compared (i.e., $\Delta_t^{\mathrm{P}}(\mathbf{x}, r) > \Delta_t^{\mathrm{P}}$).

7.1.8 Metrics of phylogenetic relative to species diversity, α_S family

The $\alpha_\mathrm{phy}(r)$ is strongly determined by the underlying pattern of neighborhood species richness. This makes it difficult to tease apart the influence of species dissimilarities and species placement if the observed $\alpha_\mathrm{phy}(r)$ is compared to null communities that randomize the locations of the individuals. Analogously to the *rISAR*, the cumulative phylogenetic mark correlation function $C_\mathrm{d}(r)$ factors out the signal of the species richness by dividing the $\alpha_\mathrm{phy}(r)$ with the $\alpha_\mathrm{S}(r)$:

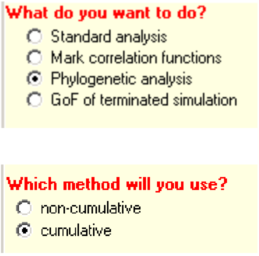
$$C_\mathrm{d}(r)=\frac{\alpha_\mathrm{phy}(r)}{\alpha_\mathrm{S}(r)},\ K_\mathrm{d}(r)=C_\mathrm{d}(r)/MPD \text{ with } MPD=\sum_{i,j=1}\delta^\mathrm{P}_{ij}f_if_j\,/\sum_{i,j=1}\delta_{ij}f_if_j$$

The $C_\mathrm{d}(r)$ is the **expected dissimilarity of two randomly selected heterospecifics in *W* that are located within distance *r***. The normalization constant is the mean pairwise dissimilarity between all heterospecific individuals in *W* (Clarke & Warwick 1998) and identical to the abundance weighted *MPD* used in quadrat-based phylogenetic analyses (de Bello et al. 2016). If the null model randomizes the dissimilarity matrix, the use of the normalized $K_\mathrm{d}(r)$ is recommended.

The cumulative phylogenetic mark correlation function at the focal species level (family F4 in the table above) **is the mean dissimilarity of all heterospecifics located within distance *r* of the typical individuals of the focal species *f*** and given by

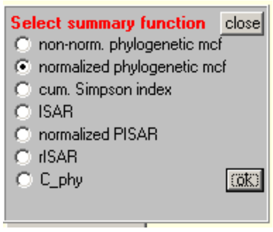
$$C_{f,\mathrm{d}}(r)=\frac{\alpha_{f,\mathrm{phy}}(r)}{\alpha_{f,\mathrm{S}}(r)},\ K_{f,\mathrm{d}}(r)=C_{f,\mathrm{d}}(r)/MPD_f \quad\text{with}\quad MPD_f=\sum_{i=1}\delta^\mathrm{P}_{fj}f_j\,/\sum_{i=1}\delta_{fj}f_j$$

The cumulative phylogenetic mark correlation function can be accessed in *Programita* by selecting “Phylogenetic analysis”, “cumulative” in the window “Which method will you use”, deselecting “With conspecifics” and “Mean of all species” (for the community level function) or “For one species” (for the individual level function) in the “Multivariate analysis” window and selecting “phylog. Simpson index” in the “summary function” window:



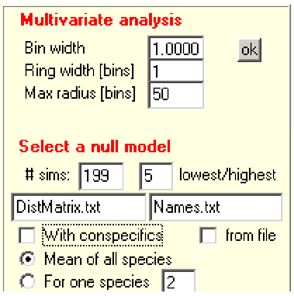
What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

Which method will you use?
○ non-cumulative
● cumulative



Select summary function

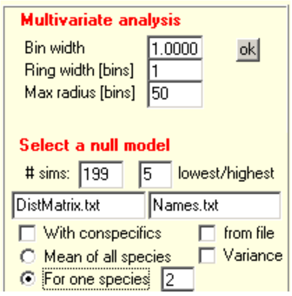
○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ cum. Simpson index
○ ISAR
○ normalized PISAR
○ rISAR
○ C_phy



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2

Community level



Multivariate analysis
Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model
# sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 2

Individual level, specify species (2 in the example)

The spatially explicit $K_{f,\mathrm{d}}(r)$ is the normalized mean pairwise dissimilarity between the typical individual of the focal species *f* and all other heterospecifics within distance *r*, whereas the $rISAR_f(r)$ is the normalized mean dissimilarity between the typical individual of the focal species *f* and all other species within distance *r*. We can therefore also derive a map $K_{t,\mathrm{d}}(\mathbf{x}, r)$ of phylogenetic neighborhood dissimilarity that considers the relative abundance f_i of the species *i* in the neighborhood of location **x**.

7.1.9 Metrics of phylogenetic relative to species diversity, β_S family

The phylogenetic mark correlation function $k_\mathrm{d}(r)$ (Shen et al. 2013; family F7) is the most important summary function to capture phylogenetic spatial structure with null models randomizing the dissimilarity matrix. It is the normalized version of the spatially-explicit phylogenetic Simpson index $\beta_\mathrm{phy}(r)$ divided by the corresponding Simpson index $\beta_\mathrm{S}(r)$ of species diversity:

$$c_\mathrm{d}(r) = \frac{\beta_\mathrm{phy}(r)}{\beta_\mathrm{S}(r)}, \; k_\mathrm{d}(r) = c_\mathrm{d}(r)/MPD \text{ with } MPD = \sum_{i,j=1} \delta_{ij}^\mathrm{P} f_i f_j \Big/ \sum_{i,j=1} \delta_{ij} f_i f_j$$

The $c_\mathrm{d}(r)$ is the **expected dissimilarity of two randomly selected heterospecifics in *W* that are distance *r* away**. The normalization constant is the mean pairwise dissimilarity between all heterospecific individuals in *W* (Clarke & Warwick 1998) and identical to the abundance weighted *MPD* used in quadrat-based phylogenetic analyses (de Bello et al. 2016).

An advantage of the phylogenetic mark-correlation function $k_\mathrm{d}(r)$ is that it focuses on the dissimilarities among neighbored heterospecifics and factors out all signals of species clustering and pairwise co-occurrence patterns. For this reason it will be most powerful if it is used in concert with null models that randomize the matrix of pairwise dissimilarities (Shen et al. 2013). Importantly, the $k_\mathrm{d}(r)$ is also normalized with the mean pairwise dissimilarity *MPD* between all heterospecific individuals in *W* which represents the information on the overall phylogenetic structure of the community in *W* relative to the species pool (i.e., phylogenetic clustering or vs. overdispersion; Webb et al. 2002). Therefore, **$k_\mathrm{d}(r)$ is independent on the overall phylogenetic structure of the community in *W*, but only driven by the small-scale spatial arrangement of individuals relative to their dissimilarities** (Shen et al. 2013, Wiegand & Moloney 2014). As a consequence, abundance phylogenetic structuring does not influence the null model assessment of $k_\mathrm{d}(r)$ together with null models that randomize the dissimilarity matrix $\delta_{ij}{}^\mathrm{P}$ or do not keep the overall species abundances f_i in *W*. However, this is a problem if $\beta_\mathrm{phy}(r)$ or $c_\mathrm{d}(r)$ would be used instead. This problem is well known in quadrat-based analyses (e.g., null models 1a vs. 1s and 1p in Hardy 2008).

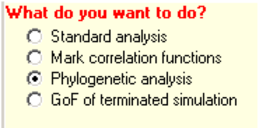
An advantage of the phylogenetic mark-correlation function is that **the value of $k_\mathrm{d}(r) = 1$ serves as dividing line between small-scale spatial phylogenetic clustering and evenness** (Shen et al. 2013; Wiegand & Moloney 2014). Therefore, if heterospecifics neighbored at distance *r* are on average more similar than the non-spatial expectation MPD = D^P/D we have spatial phylogenetic clustering (i.e., $k_\mathrm{d}(r) < 1$) and if they are more dissimilar than expected we have phylogenetic overdispersion ($k_\mathrm{d}(r) > 1$) (Shen et al. 2013). We also find $k_\mathrm{d}(r) = 1$ in two limiting cases: if the local community is not spatially structured and therefore does not show species turnover [i.e., $g_{ij}(r)/g(r) = 1$] because in this case $\beta_\mathrm{S}(r)$ collapses to *D* and $\beta_\mathrm{phy}(r)$ collapses to D^P, and if all heterospecific dissimilarities are the same.

The non-normalized phylogenetic mark correlation function at the focal species level (family F8 in the table above) **is the mean dissimilarity of all heterospecifics located at distance *r* of the typical individuals of the focal species *f*** and given by

$$k_{f,\mathrm{d}}(r) = \frac{\beta_{f,\mathrm{phy}}(r)}{\beta_{f,\mathrm{S}}(r)}, \text{ can be normalized with } MPD_f = \sum_{i=1} \delta_{fj}^\mathrm{P} f_j \Big/ \sum_{i=1} \delta_{fj} f_j$$

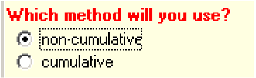
Multivariate metrics of phylogenetic relative to species diversity

The phylogenetic mark correlation function can be accessed in *Programita* by selecting "Phylogenetic analysis", "non-cumulative" in the window "Which method will you use", deselecting "With conspecifics" and "Mean of all species" (for the community level function) or "For one species" (for the individual level function) in the "Multivariate analysis" window and selecting "phylog. Simpson index" in the "summary function" window:



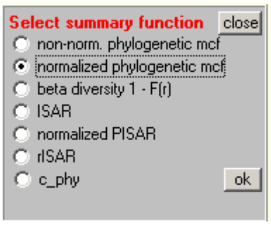
What do you want to do?

○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation



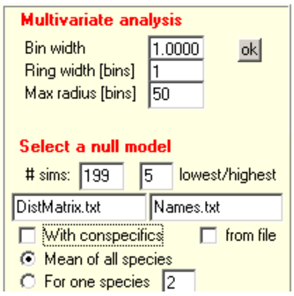
Which method will you use?

● non-cumulative
○ cumulative



Select summary function

○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy



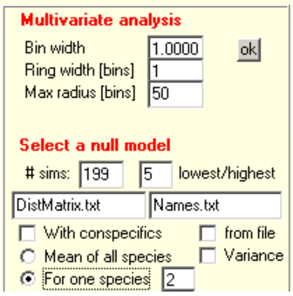
Multivariate analysis

Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model

sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2

Community level



Multivariate analysis

Bin width 1.0000
Ring width [bins] 1
Max radius [bins] 50

Select a null model

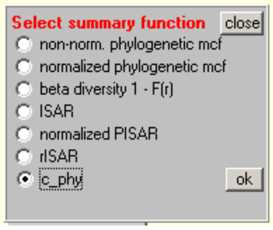
sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 2

Individual level, specify species (2 in the example)

7.1.10 The phylogenetic co-occurrence function

The non-normalized phylogenetic mark correlation function $c_d(r)$ can be estimated as

$$c_d(r) = \frac{\beta_{phy}(r)}{\beta_S(r)} = \frac{\sum_{i=1}^{S}\sum_{j=1}^{S}\delta_{ij}^{P} p_{ij}(r)}{\sum_{i=1}^{S}\sum_{j=1}^{S}\delta_{ij} p_{ij}(r)},$$



Select summary function

○ non-norm. phylogenetic mcf
○ normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
● c_phy

with $p_{ij}(r) = f_i f_j\, g_{ij}(r)/g(r)$ being the mark connection functions of the i-j species pair. Of special interest is which spatial scales and phylogenetic depths produce overall patterns of spatial phylogenetic clustering or overdispersion. Following an idea of Parmentier et al. (2014), Wiegand et al. (2017) introduced the phylogenetic co-occurrence function $c_{\mathrm{phy}}(r, I)$ that describes the spatial co-occurrence of heterospecific individuals with dissimilarities within a given phylogenetic interval $I = (\delta_{\min}^P, \delta_{\max}^P]$. The $c_{\mathrm{phy}}(r, I)$ is defined as the **probability that two heterospecifics distance *r* apart have dissimilarities within a given phylogenetic interval *I*:**

$$c_{\mathrm{phy}}(r, I) = \sum\nolimits_{i,j} \mathbf{1}[\delta_{ij}^{\mathrm{P}} \in I]\, p_{ij}(r) \,/ \sum\nolimits_{i,j} \delta_{ij}\, p_{ij}(r)\,,$$

where the indicator function **1**[.] has value of one if the argument is true and zero otherwise. Consequently, if small contagious phylogenetic intervals $I_1, I_2, .., I_n$ with midpoints δ_k^{P} cover the entire dissimilarity range we find $\Sigma_k\, c_{\mathrm{phy}}(r, I_k) = 1$ and

$$c_d(r) \approx \sum\nolimits_k \delta_k^P\, c_{\mathrm{phy}}(r, I_k)\,.$$

Thus, the **family $c_{\text{phy}}(r, I_k)$ of phylogenetic co-occurrence functions decomposes the community co-occurrence patterns at different spatial scales *r* and phylogenetic scales *I* (depths).** This allows investigating which spatial scales and phylogenetic depths produce overall patterns of spatial phylogenetic clustering or overdispersion (Parmentier et al. 2014).

The corresponding cumulative phylogenetic co-occurrence function $C_{\text{phy}}(r, I)$ is defined as the **probability that two heterospecifics within distance *r* have dissimilarities within a given phylogenetic interval *I*:**

$$C_{\text{phy}}(r, I) = \sum\nolimits_{i,j} \mathbf{1}[\delta_{ij}^{\text{P}} \in I] P_{ij}(r) / \sum\nolimits_{i,j} \delta_{ij} P_{ij}(r)$$

with $p_{ij}(r) = f_i f_j K_{ij}(r)/K(r)$ being the cumulative mark connection function of species pair i-j.

7.1.11 Multivariate data types that can be analyzed

Programita allow a variety of analysis of multivariate point patterns based on the summary functions described above and their extensions:

- “**univariate**” average analysis where the individuals taken as focal individuals [i.e., index *i* in $g_{ij}(r)$] are stemming from the same multivariate pattern as the counted individuals [i.e., index *j* in $g_{ij}(r)$]
- “**bivariate analysis**” based on two multivariate patterns where individuals of the first multivariate pattern are only used as focal individuals (i.e., index *i*) and individuals of the second multivariate patter are only used as counted individuals (i.e., index *j*). An example is the mean species richness of small trees around large trees of a given focal species.
- “**individual**” analyses where only individuals of one focal species *f* of the first multivariate pattern are used, but all individuals of the first or second multivariate pattern (i.e., index *j*) are counted for uni- and bivariate analyses, respectively (Families F2, F4, F8 in the table above). This yields for example the **individual species area relationship** $ISAR_f(r)$ (Wiegand et al. 2007a) that estimates the mean number of species within distance *r* of individuals of focal species *f*.
- analysis with a **qualitative mark** based on one multivariate pattern that carries an additional qualitative mark. This analysis views the entire community (count pattern with index *i*) from the viewpoint of points with the first qualitative make (univariate) and the second qualitative mark (bivariate). Examples are the pattern of all surviving vs. surviving large trees in a forest. An example of such an analysis is assessment the mean species richness of all larger trees at distance *r* of surviving large trees in a forest.
- **Trivariate analysis**. In this case we have two multivariate patterns and a qualitative mark in the second pattern (e.g., large trees: focal pattern *f* and small trees: counted pattern *j* that carries a qualitative mark). An example of such an analysis is the probability of mortality of small trees at distance *r* of large trees.

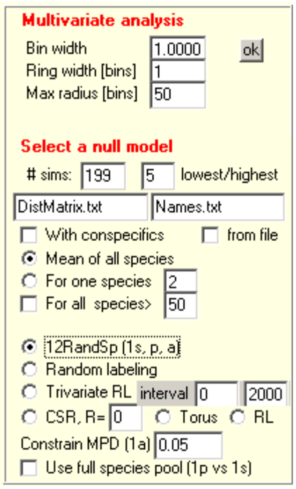
7.1.12 Null models and null communities

Multivariate analyses based on a dissimilarity matrix allow for three different types of null models:

1. The **dissimilarity matrix is randomized** in a way that spatial phylogenetic structure disappears and the summary functions $\overline{rISAR}(r)$ and $k_d(r)$ yield expectations of one for all distances *r*.
2. The **spatial locations of the individuals of the community are randomized** in a way that certain spatial structures in the data are maintained but others are randomized. For example, the toroidal shift null community maintains the clustering of individual species but removes potential spatial associations among species and potential associations of species to habitat. **The null community approach is of special interest for questions of community assembly.** Programita offers a few simple null community models, or alternatively, for more specific null communities the data files can be generated separately to be inputted into *Programita* via the "from file" option. See e.g., Wang et al. (2013, 2015) and Shen et al. (2009) for examples of the null community approach.
3. For conducting **individual** analyses, e.g., with the individual species area relationship *ISAR* or the *rISAR* function, a null model is needed for the focal species *f* that generates pairwise independence among the focal species *f* and all other species in the community. This is implemented in *Programita* via the "from file" option where the files for the null model of the focal species are generated separately and must be inputted.

Randomization of dissimilarity matrix

The **species shuffling null model** "**12RandomSp**" uses the Matrix of distances between species s1 and s2 (i.e., Matrix[s1, s2]), generates a random permutation of the vector with the species names (perm[s]) and creates a new distance matrix Matrix[perm[s1], perm[s2]] which arises by randomly shuffling the names of the species names. The **species shuffling null model** has a number of options:



Multivariate analysis

Bin width 1.0000 ok
Ring width [bins] 1
Max radius [bins] 50

Select a null model

sims: 199 5 lowest/highest
DistMatrix.txt Names.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

● 12RandSp (1s, p, a)
○ Random labeling
○ Trivariate RL interval 0 2000
○ CSR, R= 0 ○ Torus ○ RL
Constrain MPD (1a) 0.05
□ Use full species pool (1p vs 1s)

You can use the full species pool of your distance matrix (check "**Use full species pool**"). If this option is disabled, *Programita* shuffles only the names of species with more than one individual in the data set used. This corresponds to null models 1s vs. 1p in Hardy (2008) with 1s being only the realized species pool and 1p the entire species pool in the matrix.

You can also constrain the null model to accept only null matrices that yield a mean pairwise phylogenetic distance (MPD) between all heterospecific individuals of the multivariate pattern which is similar to the observed one (**Constrain MPD**). This corresponds to null model 1a in Hardy (2008).

With MPD_{obs} and MPD_{null} being the observed and null values of MPD, you can set a threshold value (e.g., *th* = 0.05) that accepts only values of MPD_{null} with $\left|(MPD_{\text{obs}} - MPD_{\text{null}})/MPD_{\text{obs}}\right| < th$. The default value is *th* = 0.05, thus the MPD of the null model are only allowed to vary by 5% from that of the observed MPD.

7.2 One multivariate pattern using a dissimilarity matrix

7.2.1 Data preparation for one multivariate pattern

For the multivariate analysis you need three data files:

1. a data file with the locations and species identity of all individuals in the community. This is an ASCII file with *.phy extension.
2. a data file with the species acronyms and the species numbers. This is an ASCII file with *.txt extension
3. a data file with the dissimilarity matrix. This is an ASCII file with *.txt extension

1) The data files for univariate analysis to detect phylogenetic (or functional) spatial structure in the fine-scale placement of individuals specify one multivariate pattern. The data files must be an ASCII file with *. phy extension and have the following format (the example data file DataType3_Habitat_12.phy):

```
0   300   0   300   12000
 87.33    98.11   1   4
170.29   126.70   1   6
 18.54   274.65   1   2
147.48   230.35   1   10
206.88   158.61   1   10
147.47   200.51   1   4
…
```

- the first line gives the dimension of the plot (300 × 300 units) in the example and the total number of points in the list (12,000 in the example)
- the first two columns are the coordinates of the points
- the third column gives the pattern (here always "1" because there is only one multivariate pattern)
- the forth column gives the species identifier (being an integer running from 1 to *S*).
- the fifth column is optional and can carry an quantitative mark to be used in specific applications.

2) The file with species numbers and species acronyms (here file Names_Habitat_12.txt) is a **tab delimited** ASCII file with the *.txt extension and the following format:

```
1   SPECI1
2   SPECI2
3   SPECI3
…
9   SPECI9
10  SPEC10
```

where the first column is the species number and the second column a **6 letter** species acronym. Please do not call this file "Names.txt".

3) The data file with distance matrix (here DistanceMatrix_Habitat_12.txt) which is **tab delimited** ASCII file with the *.txt extension and the following format:

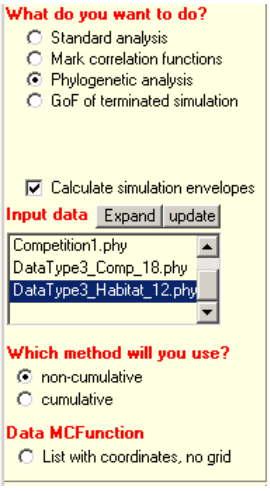
```
SPECI1  SPECI2  0.7587
SPECI1  SPECI3  0.9752
SPECI1  SPECI4  0.6250
SPECI1  SPECI5  0.4814
SPECI1  SPECI6  0.0504
SPECI1  SPECI7  0.1111
```

where the first two columns are the six letter species acronyms of the species pair and the third column is the distance between the two species. Do not call this file “DistMatrix.txt”. **Note that this file must be tab delimited, that the species acronyms in the distance matrix and the species list must exactly match, and the species number in the *.phy data file must match with that of the species list**. The file temp_MissMatch.txt indicates if a species acronym in the distance matrix file is not in the file with the species list.

7.2.2 One multivariate pattern and randomization of the dissimilarity matrix

The **univariate average analysis** can be accessed with the following sequence of actions:

1. Select “**Phylogenetic analysis**” in What do you want to do?
2. Highlight data file you want to analyze in Input data. The example file “DataType3_Habitat_12.phy” is a simulated data set where 12,000 individuals of 10 species are distributed in a 300 × 300m plot. It belongs to the habitat association communities of Shen et al. (2013: their Fig. 1c) and describes associations to a periodic habitat in x-direction. The community contains phylogenetic spatial structure caused by habitat association (habitat filtering) because niche differences between two species were highly correlated with their phylogenetic relatedness.
3. Click “**List with coordinates, no grid**” in MCFunction



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

☑ Calculate simulation envelopes

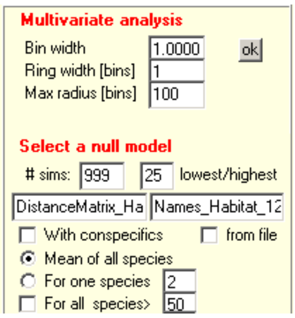
Input data Expand update

Competition1.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
○ List with coordinates, no grid

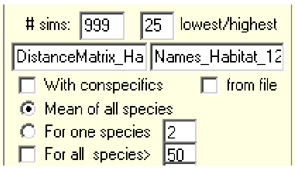
4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting “**cumulative**” in Which method will you use. However, this somewhat slows down the estimation and there the better option for estimating the cumulative index is by using the **Replicate** option
5. Provide in the window Multivariate analysis the bin width in data units, an appropriate ring width (use ring width of 1 if the analysis takes long time and then the **Replicate** option to change the ring width), and a maximal distance *r* of the analysis.



Multivariate analysis
Bin width 1.0000 ok
Ring width [bins] 1
Max radius [bins] 100

Select a null model
# sims: 999 25 lowest/highest
DistanceMatrix_Ha Names_Habitat_12
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

6. Provide the file with species numbers and names (here file “Names_Habitat_12.txt”)



# sims: 999 | 25 lowest/highest

DistanceMatrix_Ha | Names_Habitat_12

□ With conspecifics □ from file

● Mean of all species

○ For one species 2

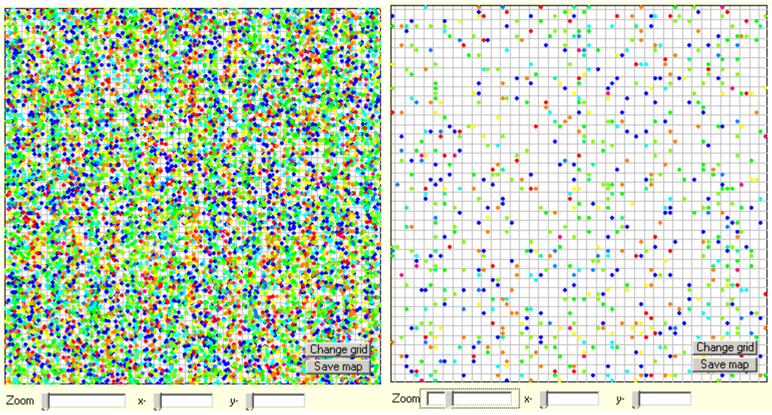
□ For all species> 50

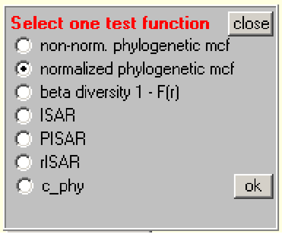
7. Provide the data file with distance matrix (here DistanceMatrix_Habitat_12.txt)

8. For the "univariate" analysis select "**Mean of all species**"

9. You can include or exclude the focal species with the check box **With conspecifics**. If you include conspecifics you obtain the phylogenetic Simpson index and if you exclude conspecifics you obtain the phylogenetic mark correlation function (see section 3.1.7.6 in Wiegand and Moloney 2014 and Shen et al. 2013).

10. Click "**Calculate Index**". *Programita* now shows you the multivariate pattern on the left with different species indicated by different colors. You can also zoom into the map to see more detail:





Select one test function close

○ non-norm. phylogenetic mcf

● normalized phylogenetic mcf

○ beta diversity 1 - F(r)

○ ISAR

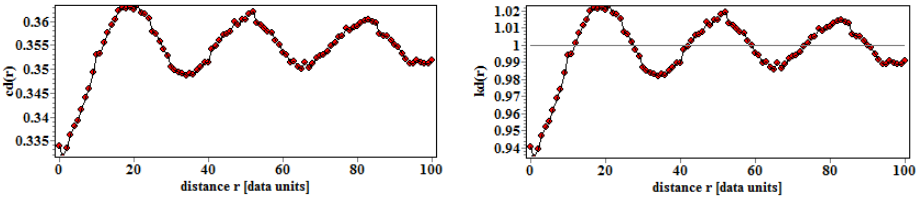
○ PISAR

○ rISAR

○ c_phy ok

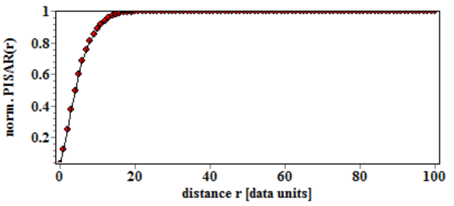
On the right you see the multivariate summary functions of the data. In window Select one test function you can select the different multivariate summary functions.

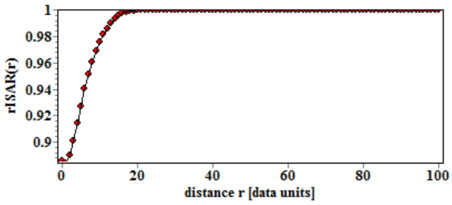
11. The results of the non-normalized phylogenetic mark correlation function $c_{\mathrm{d}}(r)$ is shown on the left, and that of the normalized $k_{\mathrm{d}}(r)$ on the right:



12. The phylogenetic mark correlation function captures the periodic habitat filtering very well and suggests phylogenetic clustering for nearby individuals and for individuals located in the repeated bands of the underlying habitat. However, individuals located in the out of phase habitats are more dissimilar than on average. Note that the oscillations are not much damped because the underlying habitat is strictly periodic in one direction.

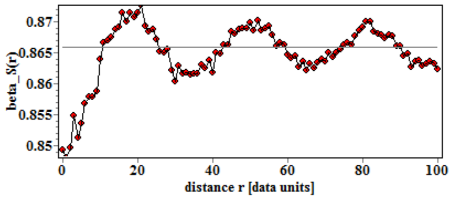
13. The normalized PISAR and *rISAR* functions capture the phylogenetic similarity of species only within one strip of the habitat

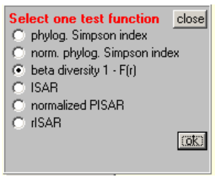




but because they are based on nearest neighbor statistics (i.e., presence of species in neighborhoods *r*) they cannot capture the periodic structure reveled by the phylogenetic mark correlation function.

14. To obtain the results for the species beta diversity [i.e., the spatially explicit Simpson index $\beta_S(r)$] you need to include the conspecifics by clicking ☑ Include conspecifics in window Multivariate analysis and click again "**Calculate Index**":





Select one test function close

○ phylog. Simpson index
○ norm. phylog. Simpson index
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR

ok

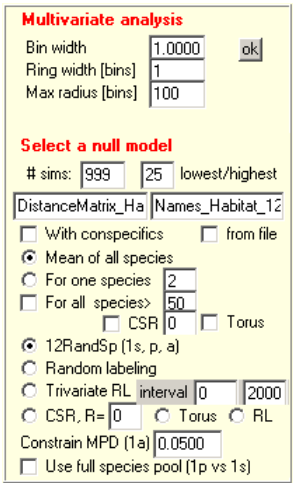
As expected by the construction of the data, the proportion heterospecifics increases up to the distance of (approximately) 20m where the habitat is out of phase. The repeated strips of habitat therefore cause the periodicity in the Simpson index $\beta_S(r)$.

15. Click "**calculate simulation envelopes**"

16. Go back to the window Multivariate analysis and select the number of simulations of the null model (999), and the rule for the simulation envelopes (25).

17. Select for the univariate analysis the species shuffling null model "**12RandomSp**" (see Shen et al. 2013).

The **species shuffling null model** uses the Matrix of distances between species s1 and s2 (i.e., Matrix[s1, s2]), generates a random permutation of the vector with the species names (perm[s]) and creates a new distance matrix Matrix[perm[s1], perm[s2]] which arises by randomly shuffling the names of the species names.



Multivariate analysis

Bin width 1.0000 ok

Ring width [bins] 1

Max radius [bins] 100

Select a null model

sims: 999 25 lowest/highest

DistanceMatrix_Ha Names_Habitat_12

□ With conspecifics □ from file

● Mean of all species

○ For one species 2

□ For all species> 50

□ CSR 0 □ Torus

● 12RandSp (1s, p, a)

○ Random labeling

○ Trivariate RL interval 0 2000

○ CSR, R= 0 ○ Torus ○ RL

Constrain MPD (1a) 0.0500

□ Use full species pool (1p vs 1s)

18. The **species shuffling null model** has a number of options:

- You can use the full species pool of your distance matrix (check "**Use full species pool**"). If this option is disabled, *Programita* shuffles only the names of species with more than one individual in the data set used. This corresponds to null models 1s vs. 1p in Hardy (2008) with 1s being only the realized species pool and 1p the entire species pool in the dissimilarity matrix.

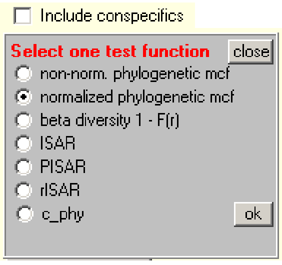
- You can also constrain the null model to accept only null matrices that yield a mean pairwise phylogenetic distance between all heterospecific individuals (MPD) which is similar to the observed one (**Constrain MPD**). This corresponds to null model 1a in Hardy (2008):

  With MPD_{obs} and MPD_{null} being the observed and null values of MPD, you can set a threshold value (e.g., $th = 0.05$) that accepts only values of MPD_{null} with $|(MPD_{\text{obs}} - MPD_{\text{null}})/MPD_{\text{obs}}| < th$.

  The default value is $th = 0.05$, thus the MPD of the null model are only allowed to vary by 5% from that of the observed MPD.

  However, if after 1000 attempts no simulated dissimilarity matrix satisfies the condition, *Programita* uses a 5 times larger threshold, after 5000 failed attempts it uses a 25 times larger threshold, and after 10000 failed attempts it uses the last randomization of the dissimilarity matrix.

19. Decide if you want to run the analysis with or without considering conspecifics: □ Include conspecifics . The beta diversity summary functions are different and the ISAR summary functions slightly different:



□ Include conspecifics

Select one test function close

○ non-norm. phylogenetic mcf

● normalized phylogenetic mcf

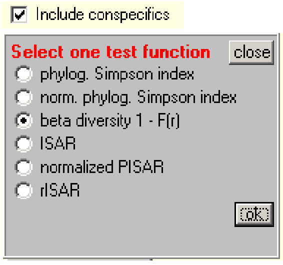
○ beta diversity 1 - F(r)

○ ISAR

○ PISAR

○ rISAR

○ c_phy ok



☑ Include conspecifics

Select one test function close

○ phylog. Simpson index

○ norm. phylog. Simpson index

● beta diversity 1 - F(r)

○ ISAR

○ normalized PISAR

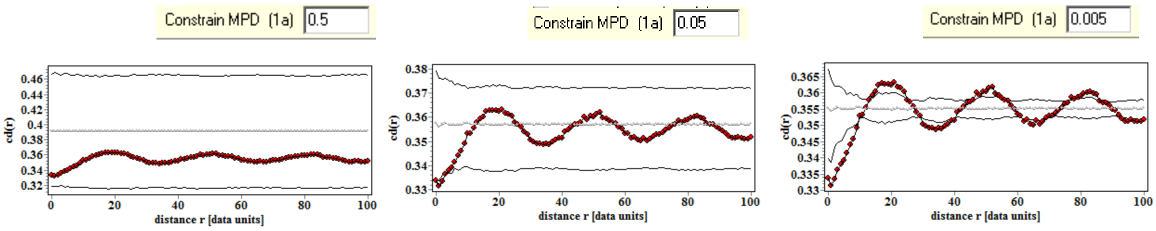
○ rISAR

ok

20. Exclude conspecifics: □ Include conspecifics

After clicking "**Calculate Index**" and running the simulations of the null model you can select with the window Select one test function among the different test functions.

21. Select “non-norm. phylogenetic mcf” to obtain the results of the **non- normalized phylogenetic mark correlation function** $c_d(r)$:

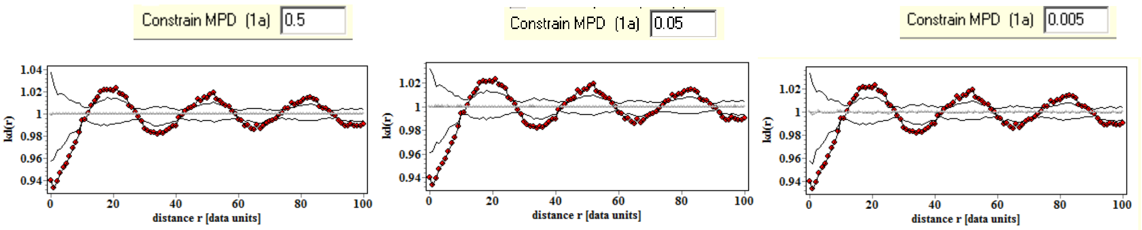

Constrain MPD (1a) 0.5

Constrain MPD (1a) 0.05

Constrain MPD (1a) 0.005

22. You notice that the simulation pointwise envelopes are very wide. This is because this summary function is not normalized and the MPD [which determines the absolute value of the $c_d(r)$] can vary widely when shuffling the names of the species in the null model. If you constrain the MPD more (default is *th* = 0.05; middle graph) you obtain narrower simulation envelopes (right) and if you constrain less the envelopes become wider (*th* = 0.5, left).

23. Because the MPD resulting from the species shuffle null model has a strong impact on the absolute values of the non-normalized phylogenetic mark correlation function $c_d(r)$ the pointwise envelopes are very wide and you cannot detect properly departures that are only due to phylogenetic structures in the small-scale placement of species. Therefore **it is recommended to use the normalized phylogenetic mark correlation function** $k_d(r)$ by selecting “normalized phylogenetic mcf”:


Constrain MPD (1a) 0.5

Constrain MPD (1a) 0.05

Constrain MPD (1a) 0.005

The simulation pointwise envelopes are now narrower and depict only phylogenetic structures in the small-scale placement of species, but are not influenced by the overall phylogenetic structure of the plot relative to the species pool or the structure of the phylogenetic tree (see Shen et al. 2013). Note that constraining the MPD has no influence on the width of the simulation envelopes of $k_d(r)$.

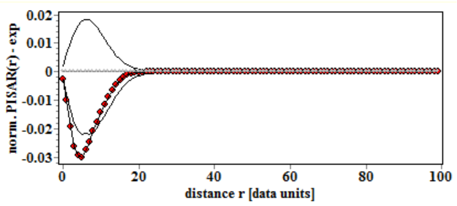
24. Note that there may be a big difference between the non-normalized phylogenetic mark correlation function $c_d(r)$ and the normalized mark correlation function $k_d(r)$.

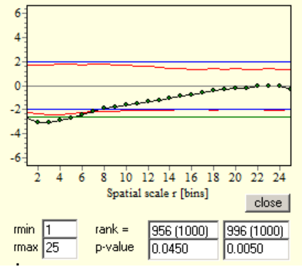
The non-normalized mark correlation function carries a signal of the overall (non-spatial) phylogenetic structure of the plot which is likely to confound the spatial phylogenetic structure you are interested in.

See Shen et al. (2013) and end of section 3.1.7.6 in Wiegand and Moloney (2014). This can be noted by the fact that the $c_d(r)$ will in general not approach at larger distances the expectation of the null model whereas the $k_d(r)$ does. As noted by Hardy 2008, this problem can be reduced by null model 1a.

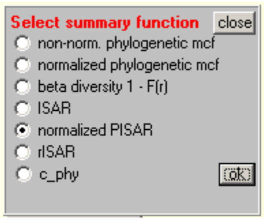
25. Select “ISAR” to obtain **the average individual-species area relationship**. However, because the null model randomizes the dissimilarity matrix, the simulations of the null model will yield always the observed ISAR.

26. Select “normalized PISAR” to obtain **the PISAR** function:



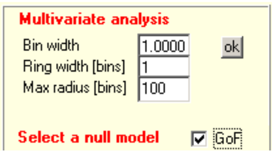


rmin 1 rank = 956 (1000) 996 (1000)
rmax 25 p-value 0.0450 0.0050


Select summary function close
- ○ non-norm. phylogenetic mcf
- ○ normalized phylogenetic mcf
- ○ beta diversity 1 - F(r)
- ○ ISAR
- ● normalized PISAR
- ○ rISAR
- ○ c_phy

ok

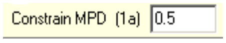
The PISAR is driven here mostly by the underlying ISAR, but if you use the GoF test ☑ GoF with the student transformation a departure from the null model can be detected at small distances.

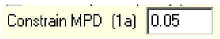

Multivariate analysis

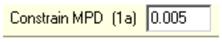
Bin width 1.0000 ok
Ring width [bins] 1
Max radius [bins] 100

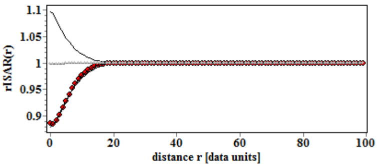
Select a null model ☑ GoF

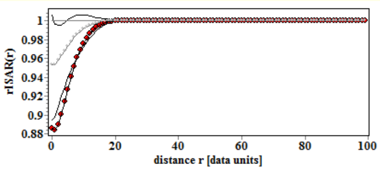
27. Select rISAR to obtain the **rIsar** function. The result is somewhat dependent on how you constrain the MPD of the null model:

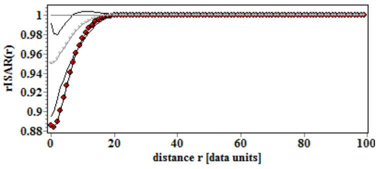

Constrain MPD (1a) 0.5


Constrain MPD (1a) 0.05


Constrain MPD (1a) 0.005





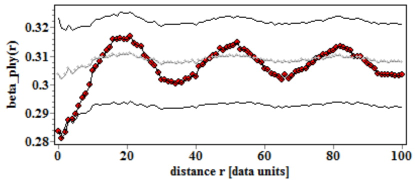


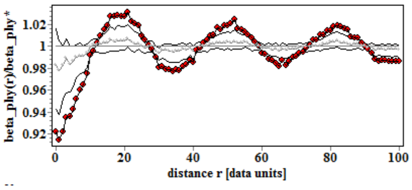
The result shows that the mean phylogenetic distance between the typical individual of the community and all other species in its neighborhoods with radius *r* is smaller than expected by the null model. Or in other words, more similar species tend to be located close to each other. However, it is clear that the phylogenetic mark correlation function captures in the present context more information than the average rISAR.

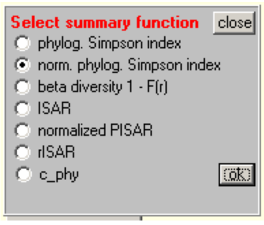
28. To obtain the phylogenetic Simpson index $\beta_{\mathrm{phy}}(r)$ enable checkbox ☑ Include conspecifics and repeat the simulation of the null model:

non-normalized $\beta_{\mathrm{phy}}(r)$

normalized $\beta_{\mathrm{phy}}(r)/\beta^{*}_{\mathrm{phy}}$






Select summary function close
- ○ phylog. Simpson index
- ● norm. phylog. Simpson index
- ○ beta diversity 1 - F(r)
- ○ ISAR
- ○ normalized PISAR
- ○ rISAR
- ○ c_phy

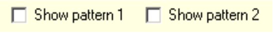
ok

Again, the normalized index shows the periodic habitat filtering quite well.

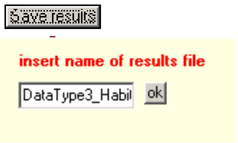
7.2.3 View results of multivariate analysis with combine replicates

You can use the "**Replicate**" option to view the results of one phylogenetic analysis and to change the estimator, for example changing the ring width or using cumulative summary functions instead of non-cumulative ones.

1. Run analysis example DataType3_Habitat12.res. To be faster disable the options "Show patterns 1 and 2". Once the simulation is finished click the "**Save results**" button and insert the name of the results file. Select as name "DataType3_Habitat_12" and two results files are generated: name.res and mcf_name_phy.rep. The "_phy" extension at the end tells the combine replicates procedure that you conducted an analysis with a dissimilarity matrix.



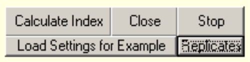
□ Show pattern 1 □ Show pattern 2

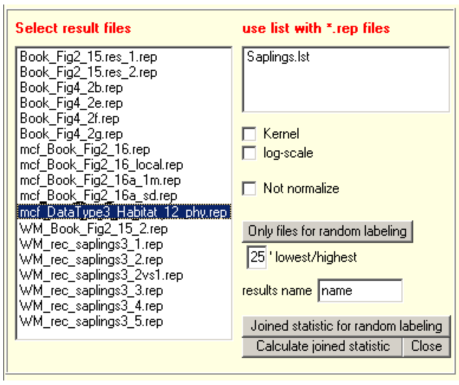


insert name of results file

DataType3_Habil

2. Click button "Replicate". A window appears where you can select the results





Select result files

Book_Fig2_15.res_1.rep
Book_Fig2_15.res_2.rep
Book_Fig4_2b.rep
Book_Fig4_2e.rep
Book_Fig4_2f.rep
Book_Fig4_2g.rep
mcf_Book_Fig2_16.rep
mcf_Book_Fig2_16_local.rep
mcf_Book_Fig2_16a_1m.rep
mcf_Book_Fig2_16a_sd.rep
mcf_DataType3_Habitat_12_phy.rep
WM_Book_Fig2_15_2.rep
WM_rec_saplings3_1.rep
WM_rec_saplings3_2.rep
WM_rec_saplings3_2vs1.rep
WM_rec_saplings3_3.rep
WM_rec_saplings3_4.rep
WM_rec_saplings3_5.rep

use list with *.rep files

Saplings.lst

□ Kernel
□ log-scale
□ Not normalize

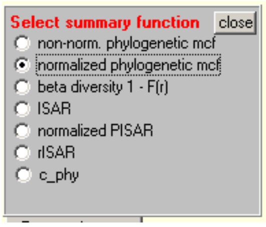
25 ' lowest/highest

results name: name

Only files for random labeling
Joined statistic for random labeling
Calculate joined statistic
Close

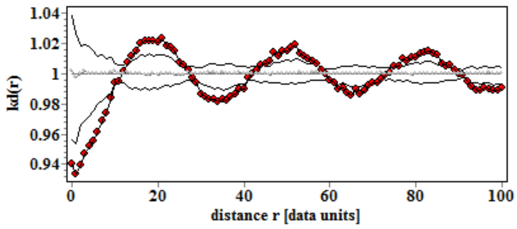
Select "mcf_DataType3_Habitat_12_phy.rep", and they click button "Calculate joined statistic". *Programita* now shows you the results of the analysis and you can select as before among the different summary functions.

3. Select the normalized phylogenetic mark correlation function:

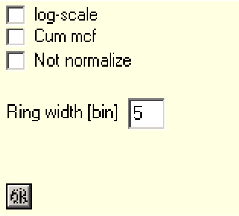


Select summary function

○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy

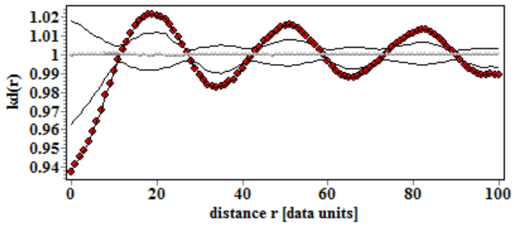


Now select a ring width of 5 units and press the "ok" button to obtain the normalized phylogenetic mark correlation function with a larger ring width. The function is much smoother:

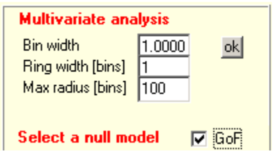


□ log-scale
□ Cum mcf
□ Not normalize

Ring width [bin] 5



4. Check the checkbox “GoF” to conduct the Goodness of Fit test and select the student transformation of the residuals



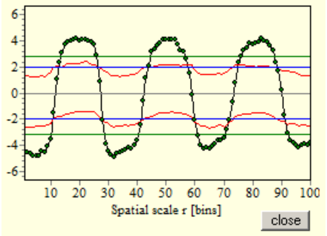
Multivariate analysis

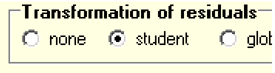
Bin width 1.0000

Ring width [bins] 1

Max radius [bins] 100

Select a null model ☑ GoF



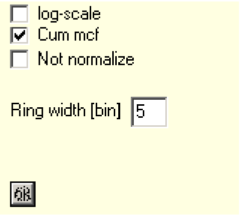


Transformation of residuals

○ none ● student ○ glot

You observe that the effect size does not decline with larger distances as you would expect because the habitat is periodic.

5. Now click “Cum mcf” to obtain the cumulative counterpart of the phylogenetic mark correlation function that yields the expected phylogenetic distance of two heterospecific individuals which are separated by a distance less than *r*, normalized with the expected phylogenetic distance c_d of two heterospecific individuals taken randomly from the plot:

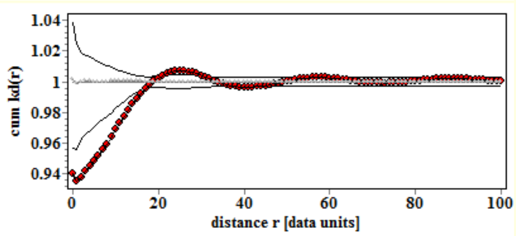


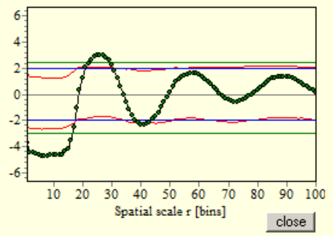
☐ log-scale

☑ Cum mcf

☐ Not normalize

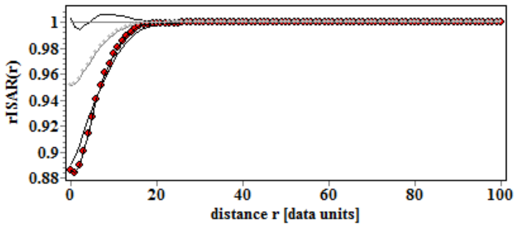
Ring width [bin] 5

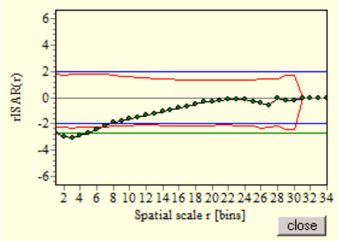




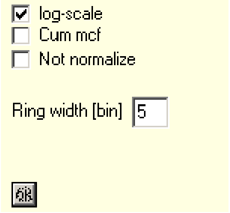
You observe that **the cumulative nature of the summary function tends to obscure the periodic nature of the species similarity induced by the periodic habitat**. Only the first phase and antiphase are detected to be significant.

Here for comparison the results of the rISAR function which is not able to reveal the oscillations because it looks only to the nearest neighbors:





6. Now click “log-scale” to view the results with a logarithmic x-axis:

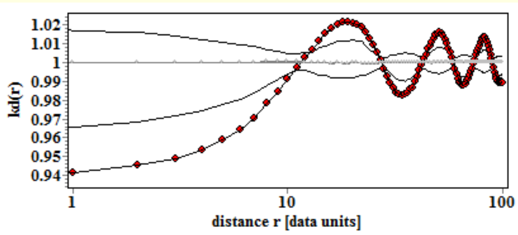


☑ log-scale

☐ Cum mcf

☐ Not normalize

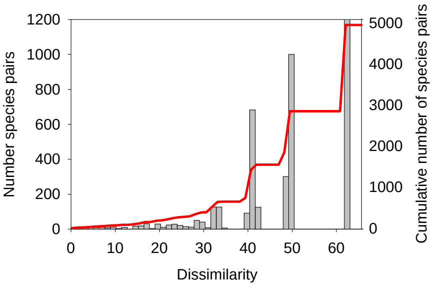
Ring width [bin] 5



7.2.4 One multivariate pattern: competition community

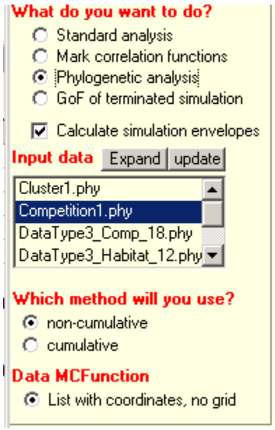
This data set is one of the examples for a community assembled by competition based on Miller et al. (2017) and analyzed in Wiegand et al. (2017). To generate the data, Wiegand et al. (2017) used the R package metricTester presented in Miller et al. (2017) to simulate phylogenies and individual-based communities within a 316 × 316 m observation window *W*.

The species dissimilarities of the corresponding phylogeny (file Dist_Competition1.txt) are continuously distributed for smaller distances (i.e., < 34), but discrete for larger distances with "clusters" around dissimilarities 41, 50 and 62.4:



To mimic competition, metricTester considered an interaction range of 20m and repeatedly removed one of two closely related individuals within this distance and replaced it by a randomly located individual from the species pool. The example data file "Competition1.phy" consists of 3843 individuals of 100 species.

The **univariate average analysis** can be accessed with the following sequence of actions:



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation
☑ Calculate simulation envelopes

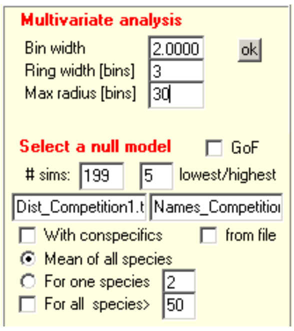
Input data

Cluster1.phy
Competition1.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy

Which method will you use?
● non-cumulative
○ cumulative

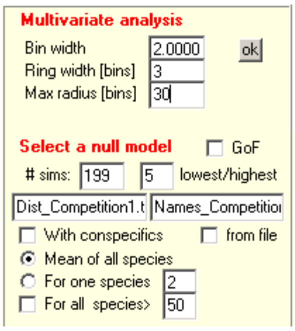
Data MCFunction
● List with coordinates, no grid

1. Select "**Phylogenetic analysis**" in What do you want to do?
2. Highlight data file you want to analyze in Input data.
   The example file "Competition1.phy" is a simulated data set where 3843 individuals of 100 species are distributed in a 316 × 316m plot. It belongs to the competition communities of Wiegand et al. (2017: their Fig. 3). The community contains phylogenetic spatial structure caused by competition of closely related species.
3. Click "**List with coordinates, no grid**" in MCFunction
4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting "**cumulative**" in Which method will you use.
   However, this somewhat slows down the estimation and the better option for estimating the cumulative index is using the **Replicate** option after the results of a non-cumulative analysis with ring width 1 are saved as *.rep and *.res file.
5. Provide in the window Multivariate analysis the bin width in data units, an appropriate ring width (use ring width of 1 if the analysis takes long time and then the **Replicate** option to change the ring width), and a maximal distance *r* of the analysis. Wiegand et al. (2017) selected a bin width of 2, a ring width of 3, and a maximal distance of 30 bins.



Multivariate analysis
Bin width 2.0000
Ring width [bins] 3
Max radius [bins] 30

Select a null model ☐ GoF
# sims: 199 5 lowest/highest
Dist_Competition1.t Names_Competitioi
☐ With conspecifics ☐ from file
● Mean of all species
○ For one species 2
☐ For all species> 50

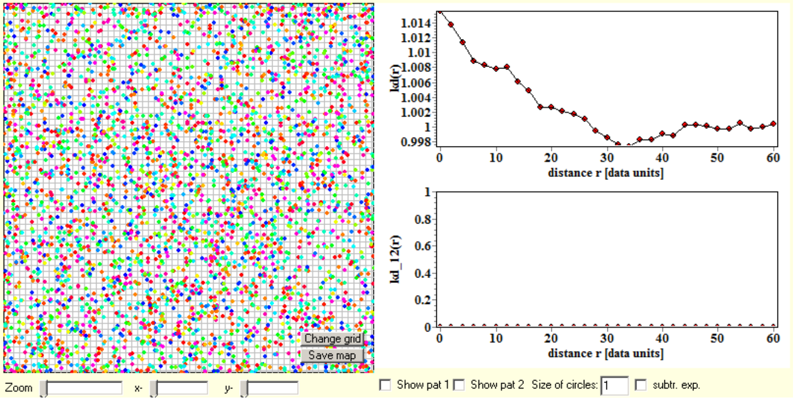

Multivariate analysis
Bin width 2.0000 ok
Ring width [bins] 3
Max radius [bins] 30
Select a null model □ GoF
# sims: 199 5 lowest/highest
Dist_Competition1.t Names_Competitio
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

6. Provide the file with species numbers and acronyms (here file "Names_Competition1.txt")
7. Provide the data file with the distance matrix (here Dist_Competition1.txt)

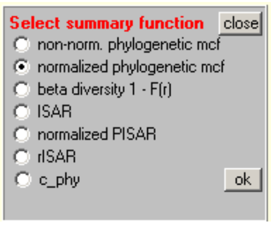
8. For the "univariate" analysis select "**Mean of all species**"

9. You can include or exclude the focal species with the check box **With conspecifics**. If you include conspecifics you obtain the phylogenetic Simpson index and if you exclude conspecifics you obtain the phylogenetic mark correlation function (see section 3.1.7.6 in Wiegand and Moloney 2014 and Shen et al. 2013).

10. Click "**Calculate Index**". *Programita* now shows you the multivariate pattern on the left with different species indicated by different colors:



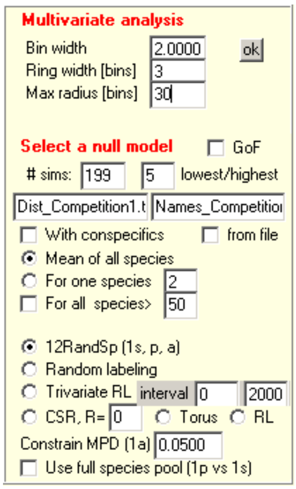
On the right you see the multivariate summary functions of the data. In window Select one test function you can select the different multivariate summary functions.


Select summary function close
○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy ok

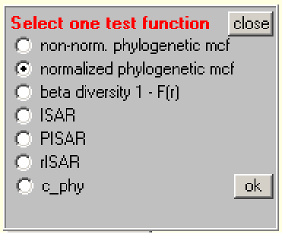
11. Click "**calculate simulation envelopes**"

12. Go back to the window Multivariate analysis and select the number of simulations of the null model (199), and the rule for the simulation envelopes (5).
13. Select for the univariate analysis the species shuffling null model "**12RandomSp**" (see Shen et al. 2013).

14. The **species shuffling null model** randomly shuffles the species names in the dissimilarity matrix. Select "**Constrain MPD (1a)**" to constrain the null model to accept only null dissimilarity matrices that yield a MPD similar to the observed one. This corresponds to null model 1a in Hardy (2008).
15. Exclude conspecifics: □ With conspecifics to obtain the phylogenetic mark correlation functions.

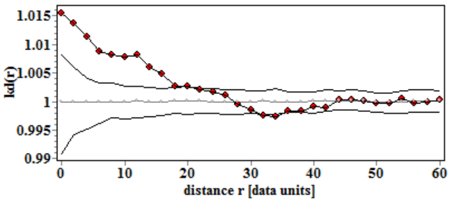

Multivariate analysis
Bin width 2.0000 ok
Ring width [bins] 3
Max radius [bins] 30
Select a null model □ GoF
# sims: 199 5 lowest/highest
Dist_Competition1.t Names_Competitio
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50
● 12RandSp (1s, p, a)
○ Random labeling
○ Trivariate RL interval 0 2000
○ CSR, R= 0 ○ Torus ○ RL
Constrain MPD (1a) 0.0500
□ Use full species pool (1p vs 1s)

16. After clicking “**Calculate Index**” and running the simulations of the null model you can select with the window Select one test function among the different test functions.



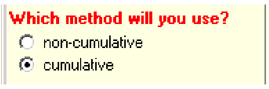
Select one test function close
○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ PISAR
○ rISAR
○ c_phy
ok

17. Select “normalized phylogenetic mcf” for results of the **normalized phylogenetic mark correlation function** $k_d(r)$:

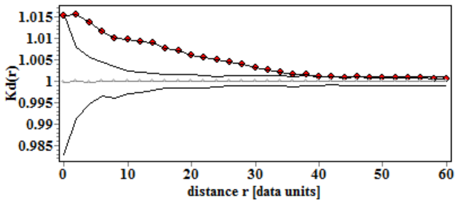


The $k_d(r)$ indicates that the expected dissimilarity of two randomly selected individuals at a fixed distance between 1 and 16m is significantly larger than expected by the randomized dissimilarity matrix. This was expected by the construction of the community. There is also a slight tendency of being more similar than expected if two individuals are some 33m away. This tendency to phylogenetic clustering arises because two neighbors B and C located at the edge of the zone of influence of a focal individual A will tend to be ecologically similar because AB and AC are dissimilar, as depicted by $k_d(r)$.

18. Select “cumulative”, and click again “Calculate index” to obtain the results of the corresponding cumulative summary function $K_d(r)$:

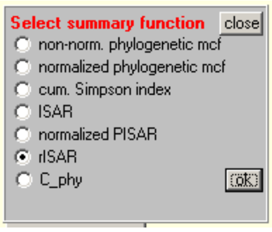


Which method will you use?
○ non-cumulative
● cumulative

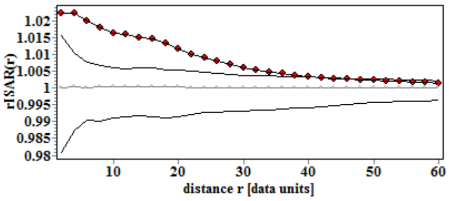


The cumulative $K_d(r)$ indicates that the expected (normalized) dissimilarity of two randomly selected individuals within some 40m is significantly larger than expected by the randomized dissimilarity matrix. However, because of the cumulative nature of the $K_d(r)$, the details shown by the $k_d(r)$ disappear.

19. Select now “rISAR” to get the average *rISAR* function that gives the expected (normalized) dissimilarity of the typical individual of the community to all other species within distance *r*:



Select summary function close
○ non-norm. phylogenetic mcf
○ normalized phylogenetic mcf
○ cum. Simpson index
○ ISAR
○ normalized PISAR
● rISAR
○ C_phy
ok

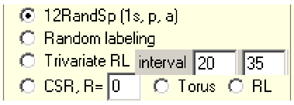


The results of the *rISAR* are very similar to that of the cumulative $K_d(r)$.

Phylogenetic co-occurrence function

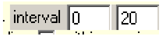
The phylogenetic co-occurrence function $c_{\mathrm{phy}}(r, I)$ describes the spatial co-occurrence of heterospecific individuals with dissimilarities within a given phylogenetic interval $I = (\delta^P_{\min}, \delta^P_{\max}]$. The $c_{\mathrm{phy}}(r, I)$ is defined as the **probability that two heterospecifics distance *r* apart have dissimilarities within a given phylogenetic interval *I***.

20. You can select the distance interval in the window "Multivariate analysis".

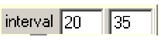


● 12RandSp (1s, p, a)
○ Random labeling
○ Trivariate RL interval 20 35
○ CSR, R= 0 ○ Torus ○ RL

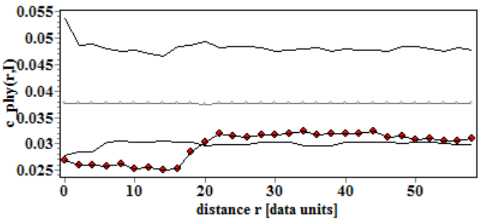
Select first intervals of small dissimilarities between 0 and 20 and 20 and 35 and run the analyses by clicking "Calculate index":

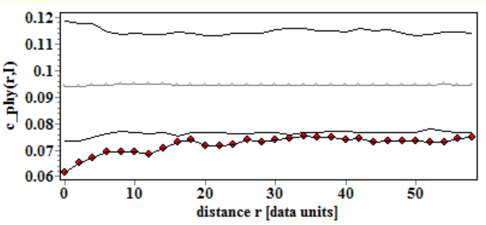


interval 0 20



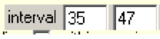
interval 20 35



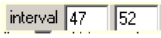


The results show that individuals with short phylogenetic distances do co-occur spatially less than expected by the randomized dissimilarity matrix. This is expected by construction of the community.

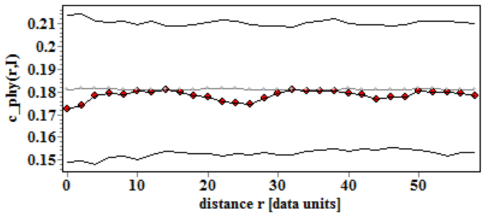
21. Select now intermediate dissimilarities between 35 and 47 and larger dissimilarities between 47 and 52:

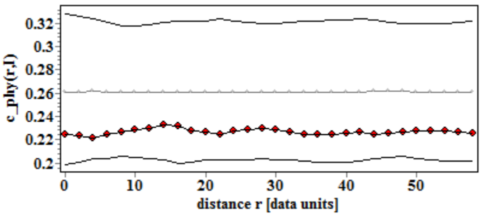


interval 35 47

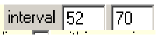


interval 47 52

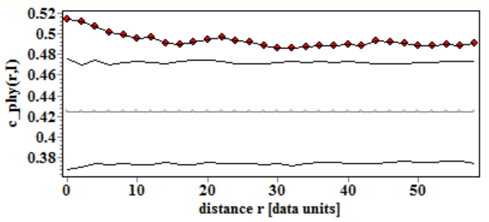




The results show that those individuals with short phylogenetic distances do not co-occur spatially different from the expectation of the randomized dissimilarity matrix. However, the many species pairs with the largest dissimilarity of 63.05 co-occur more frequently than expected by the randomized dissimilarity matrix:



interval 52 70



7.3 Individual multivariate analysis using a dissimilarity matrix

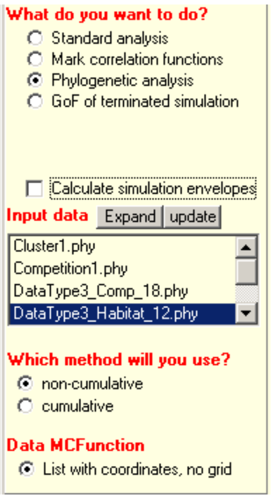
Instead of using summary functions that describe the community average, you can also use "**individual**" summary functions such as the individual species area relationship *ISAR* (Wiegand et al. 2007) or the *rISAR* function (Wang et al. 2016) that allow you to conduct the analysis from the viewpoint of the individuals of a given "**focal**" species as focal points. *Programita* uses in the individual analysis only the individuals of the focal species as focal pattern but counts individuals of all other species in the neighborhood of the focal individuals.

The summary functions of the individual analyses belong to families F2, F4, F8 in the framework of the table at the beginning of the chapter.

7.3.1 Individual analysis of one multivariate pattern

The **univariate individual analysis** can be accessed with the same sequence of actions as the standard average analysis:

1. Select "**Phylogenetic analysis**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. The example file "DataType3_Habitat_12.phy" is a simulated data set of the habitat association communities of Shen et al. (2013: their Fig. 1c) and describes habitat filtering with respect to a periodic habitat in x-direction.
3. Click "**List with coordinates, no grid**" in MCFunction
4. Provide in the window Multivariate analysis the bin width in data units (1), an appropriate ring width (use ring width of 1 if the analysis takes long time and then the **Replicate** option to change the ring width), and a maximal distance *r* of the analysis (100).



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

□ Calculate simulation envelopes

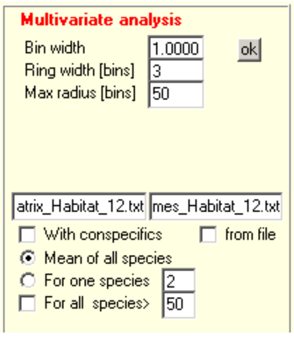
Input data

Cluster1.phy
Competition1.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
● List with coordinates, no grid

5. Provide the data file with species numbers and names (here file "sim_12.txt")
6. Provide the data file with distance matrix (here "DistanceMatrix_Habitat_12.txt")

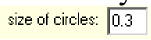


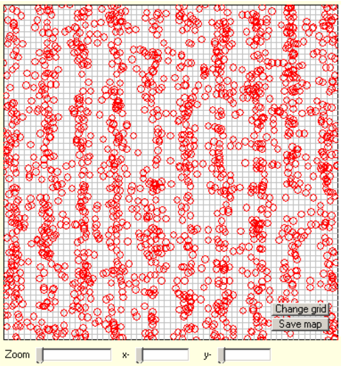
Multivariate analysis

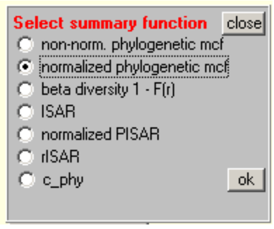
Bin width	1.0000	ok
Ring width [bins]	3	
Max radius [bins]	50	

atrix_Habitat_12.txt mes_Habitat_12.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

7. For the individual analysis select "**One focal species**" and select the species number of the focal species (6 in the example)
8. You can include or exclude the focal species from the count pattern *m* with the check box **With conspecifics**. If you include conspecifics you obtain the individual phylogenetic Simpson index and if you exclude conspecifics you obtain the individual phylogenetic mark correlation function (see section 3.1.7.6 in Wiegand and Moloney 2014 and Shen et al. 2013).

9. Click “**Calculate Index**”. *Programita* now shows you the pattern. The banded pattern of species 6 caused by the periodic habitat can be seen best with the option  size of circles: 0.3 :

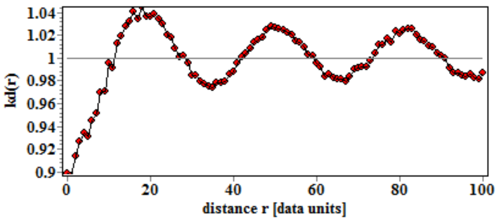



Select summary function

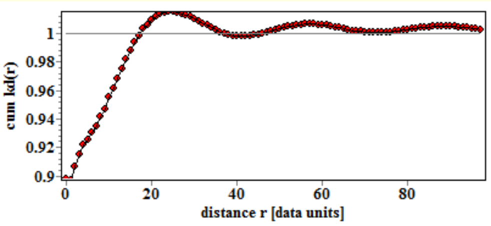
○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy

In window Select one test function you can select the multivariate summary functions of the data, for example the individual phylogenetic mark correlation function for species 6:

non-cumulative

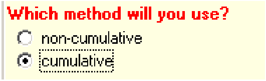


cumulative



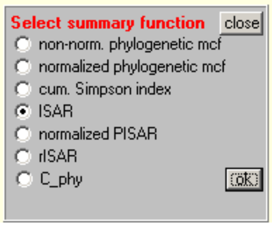
The normalized individual phylogenetic mark correlation function $k_{d,f}(r)$ yields the expected phylogenetic distance of the typical individual of the focal species *f* and a heterospecific individuals located at distance *r* and is normalized with the expected phylogenetic distance $c_{d,f}$ of the typical individual of the focal species *f* to a heterospecific individuals taken randomly from the plot.

Up to distance of 10m the species is mostly surrounded by more similar species (habitat filtering) and at distance 20m by more dissimilar species as imposed by the periodic habitat. The cumulative version somewhat obscures the effects of the periodic habitat. It can be obtained by selecting “cumulative” in the window “Which method will you use?” and clicking “Calculate index”:

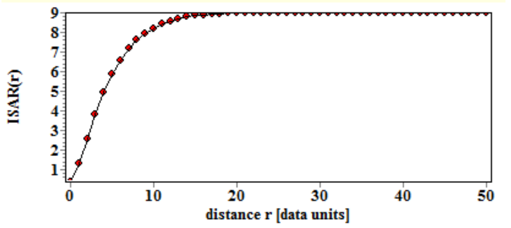

Which method will you use?

○ non-cumulative
● cumulative

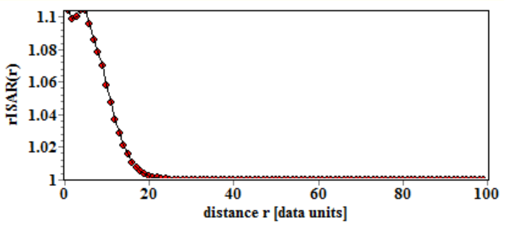
10. The *ISAR* of focal species 6 shows that neighborhoods of say 20m contain all other 9 species within the community:



○ non-norm. phylogenetic mcf
○ normalized phylogenetic mcf
○ cum. Simpson index
● ISAR
○ normalized PISAR
○ rISAR
○ C_phy



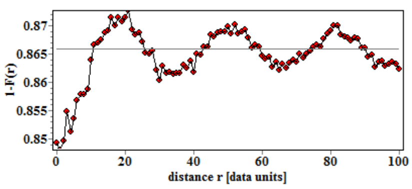
11. The *rISAR* functions yields the mean phylogenetic distance between the typical individuals of focal species 6 and all other species in its neighborhoods with radius *r*, normalized with the mean phylogenetic distance between the typical individuals of focal species 6 and all species present in the community. The result shows that the focal species 6 is surrounded at small neighborhoods by more dissimilar species and saturates, as expected, at distances of 20m to the non-spatial expectation (of one) because each individual of the focal species has all other species within 20m neighborhoods:



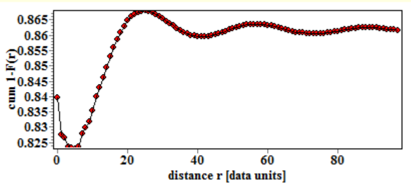
12. To assess the individual species beta diversity from the viewpoint of the focal species *f* enable the option "With conspecifics", click "**Calculate Index**", and select "beta diversity 1 - F(r)":

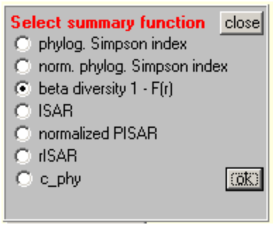
☑ Include conspecifics

non-cumulative:



cumulative

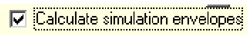




○ phylog. Simpson index
○ norm. phylog. Simpson index
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy

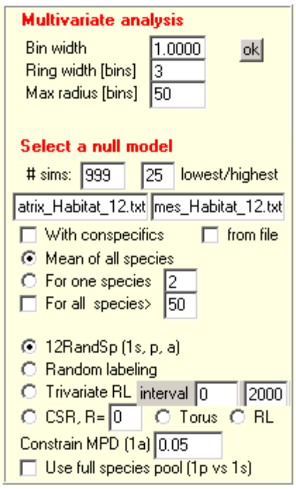
13. The probability $\beta_{S,f}(r)$ that an individual distance *r* apart from the typical individual of species *f* = 6 is heterospecific yields at small distances of say 3m a value of 85% and reaches at distance of 21m a maximum of 87%. Note that this index can be interpreted as an (inverse) index of local dominance. The cumulative version of the index $\alpha_{S,f}(r)$ is shown on the right.

14. Click “**calculate simulation envelopes**”



☑ Calculate simulation envelopes

15. Go back to the window Multivariate analysis and select the number of simulations of the null model (999), and the rule for the simulation envelopes (25). Select a ring width of *dr* = 3.



Multivariate analysis

Bin width 1.0000 ok

Ring width [bins] 3

Max radius [bins] 50

Select a null model

sims: 999 25 lowest/highest

atrix_Habitat_12.txt mes_Habitat_12.txt

□ With conspecifics □ from file

● Mean of all species

○ For one species 2

□ For all species> 50

● 12RandSp (1s, p, a)

○ Random labeling

○ Trivariate RL interval 0 2000

○ CSR, R= 0 ○ Torus ○ RL

Constrain MPD (1a) 0.05

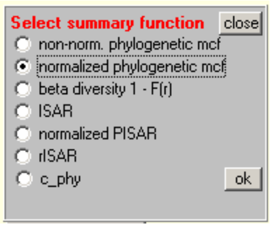
□ Use full species pool (1p vs 1s)

16. Select for the univariate analysis the species shuffling null model “**12RandomSp**” (see Shen et al. 2013). The **species shuffling null model** randomly shuffles the names of the species in the dissimilarity Matrix.

17. Disable the option “With conspecifics” □ With conspecifics

18. After clicking “**Calculate Index**” and running the simulations of the null model you can select in the window Select one test function among the different test functions.

19. Select “normalized phylogenetic mcf” to obtain the individual phylogenetic mark correlation function $k_{d,f}(r)$ that yields the expected phylogenetic distance of the typical individual of the focal species *f* and a heterospecific individuals located at distance *r*, normalized with the non-spatial expectation:



Select summary function close

○ non-norm. phylogenetic mcf

● normalized phylogenetic mcf

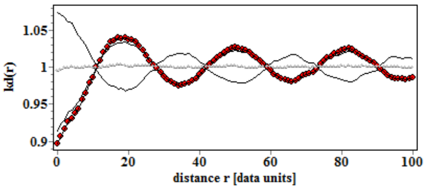
○ beta diversity 1 - F(r)

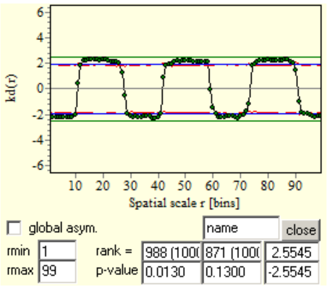
○ ISAR

○ normalized PISAR

○ rISAR

○ c_phy ok





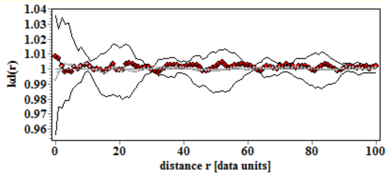
□ global asym.

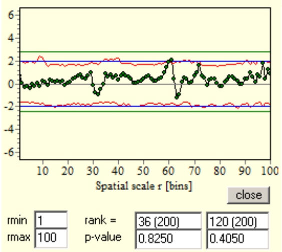
rmin 1 rank = 988 (100(871 (100(2.5545

rmax 99 p-value 0.0130 0.1300 -2.5545

The analysis shows that the focal species shows just non-significant associations to its phylogenetic neighborhoods as shown by the global envelope test.

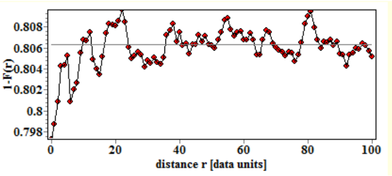
20. You can repeat the analysis also for other focal species, for example species 5 does not show departures from the null model:





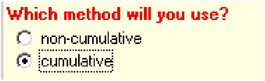
rmin 1 rank = 36 (200) 120 (200)

rmax 100 p-value 0.8250 0.4050



Note that species 5 does not show local dominance; the probability $\beta_{S,f}(r)$ that an individual distance *r* apart from the typical individual of species *f* = 5 is heterospecific varies only between value of 79.8% and 0.81%.

21. To compare the local dominance of species 5 and species 6 you can select the cumulative index $\alpha_{S,f}(r)$ by ☑ Include conspecifics and clicking "cumulative" in Which method will you use:

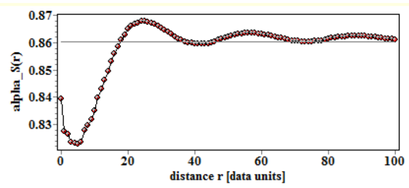


Which method will you use?

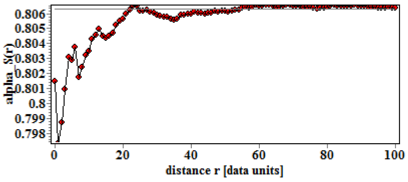
○ non-cumulative

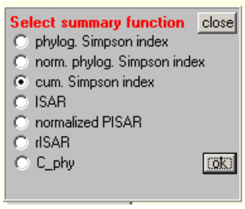
● cumulative

species 6, cumulative



species 5, cumulative





Select summary function

close

○ phylog. Simpson index

○ norm. phylog. Simpson index

● cum. Simpson index

○ ISAR

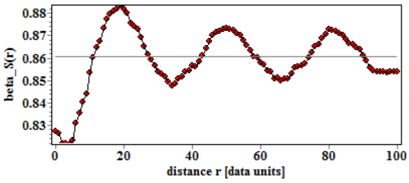
○ normalized PISAR

○ rISAR

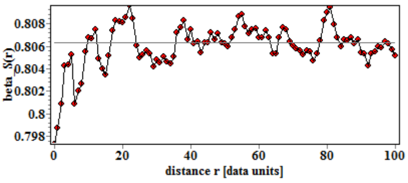
○ C_phy

ok

species 6, non cumulative



species 5, non cumulative

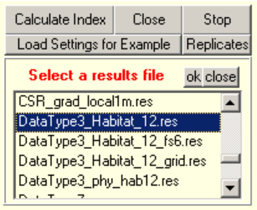


The cumulative index $\alpha_{S,f}(r)$ yields the proportion of heterospecifics within distance *r* of the typical individual of the focal species *f*. Thus, the index **1 - $\alpha_{S,f}(r)$ has the direct interpretation of an index of local dominance**; it is the proportion of conspecifics within distance *r* of the typical individual of the focal species. The grey line in the graphs indicates the value of the Simpson index (i.e., the non-spatial expectation).

Species 6 has a banded pattern that follows the periodic habitat and shows strong variation in local dominance ranging from 82.5% heterospecifics up to 87% heterospecifics. In contrast, focal species 5 varies only between 79.8% and 80.6% heterospecifics. This explains why this species does not show patterns of phylogenetic association to its neighbors.

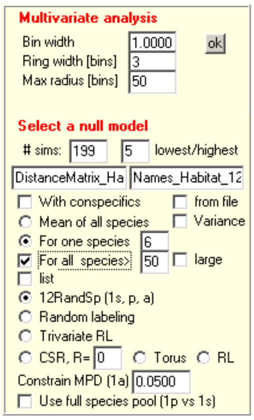
7.3.2 Series of individual analysis of one multivariate pattern

When running individual analyses one is in general interested to obtain results for all species in the community to understand variability in individual responses. Therefore, Programita offers the convenient possibility to run individual analyses of all focal species that have a minimum number of individuals. *Programita* outputs for this series analysis an additional file with a results summary. To run the series of analyses follow the following steps:



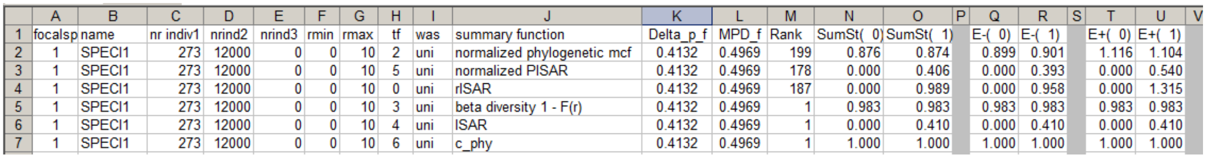
CSR_grad_local1m.res
DataType3_Habitat_12.res
DataType3_Habitat_12_fs6.res
DataType3_Habitat_12_grid.res
DataType3_phy_hab12.res

1. Load settings from file DataType3_Habitat12.res using the "**Load Settings for Example**" option.
2. Highlight the file DataType3_Habitat12.res and click the small ok button
3. *Programita* now loads all settings from this analysis. Change the maximal distance to 50, the ring width to 3, the number of simulations to 199, and the rule for the simulation envelopes to the 5th lowest and highest, otherwise the *.rep results files will become quite large. To save the *.rep file, disable the checkbox "large": ☑ large

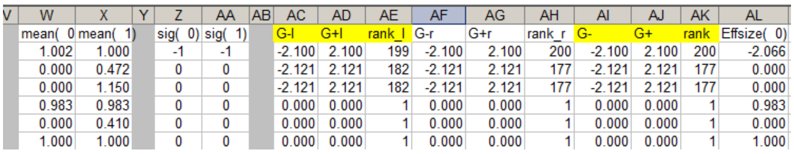


Bin width 1.0000
Ring width [bins] 3
Max radius [bins] 50
# sims: 199 5 lowest/highest
DistanceMatrix_Ha Names_Habitat_12
□ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 6
☑ For all species> 50 □ large
□ list
● 12RandSp (1s, p, a)
○ Random labeling
○ Trivariate RL
○ CSR, R= 0 ○ Torus ○ RL
Constrain MPD (1a) 0.0500
□ Use full species pool (1p vs 1s)

4. Enable the checkbox "**Run all focal species**" and provide the minimal number of individuals of the focal species (default value is 50).
5. Select a summary function in window Select one test function, the *.res results file will use this test function.
6. Click "**Calculate Index**" and *Programita* runs the individual analyses of all focal species with more than 50 individuals. To speed up to estimation, disable the options that plot the focal pattern after each simulation of the null model:
   □ Show pat 1 □ Show pat 2 Size of circles: 0.3
7. *Programita* generates results files "name_fsp_nr_phy.res" and "mcf_name_fsp_nr_phy.rep" where "name" is the file name (here "DataType3_Habitat_12") and "nr" the number of the focal species. For example, the first results file in the series for species 1 is DataType3_Habitat_12_fsp_1_phy.res.
8. *Programita* also outputs a file with a results summary named "name.txt" where the name is the name of the data file (here "DataType3_Habitat_12"):



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	focalsp	name	nr indiv1	nrind2	nrind3	rmin	rmax	tf	was	summary function	Delta_p_f	MPD_f	Rank	SumSt(0)	SumSt(1)		E-(0)	E-(1)		E+(0)	E+(1)	
2	1	SPECI1	273	12000	0	0	10	2	uni	normalized phylogenetic mcf	0.4132	0.4969	199	0.876	0.874		0.899	0.901		1.116	1.104	
3	1	SPECI1	273	12000	0	0	10	5	uni	normalized PISAR	0.4132	0.4969	178	0.000	0.406		0.000	0.393		0.000	0.540	
4	1	SPECI1	273	12000	0	0	10	0	uni	rISAR	0.4132	0.4969	187	0.000	0.989		0.000	0.958		0.000	1.315	
5	1	SPECI1	273	12000	0	0	10	3	uni	beta diversity 1 - F(r)	0.4132	0.4969	1	0.983	0.983		0.983	0.983		0.983	0.983	
6	1	SPECI1	273	12000	0	0	10	4	uni	ISAR	0.4132	0.4969	1	0.000	0.410		0.000	0.410		0.000	0.410	
7	1	SPECI1	273	12000	0	0	10	6	uni	c_phy	0.4132	0.4969	1	1.000	1.000		1.000	1.000		1.000	1.000	



V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
	mean(0	mean(1)		sig(0)	sig(1)		G-l	G+l	rank_l	G-r	G+r	rank_r	G-	G+	rank	Effsize(0)
	1.002	1.000		-1	-1		-2.100	2.100	199	-2.100	2.100	200	-2.100	2.100	200	-2.066
	0.000	0.472		0	0		-2.121	2.121	182	-2.121	2.121	177	-2.121	2.121	177	0.000
	0.000	1.150		0	0		-2.121	2.121	182	-2.121	2.121	177	-2.121	2.121	177	0.000
	0.983	0.983		0	0		0.000	0.000	1	0.000	0.000	1	0.000	0.000	1	0.983
	0.000	0.410		0	0		0.000	0.000	1	0.000	0.000	1	0.000	0.000	1	0.000
	1.000	1.000		0	0		0.000	0.000	1	0.000	0.000	1	0.000	0.000	1	1.000

9. The results summary provides information on:

- focalsp: the number of the focal species
- name: the acronym of the focal species
- nr indiv1: number of individuals of the focal species
- nrind2: number of individuals of the first multivariate pattern
- nrind3: number of individuals of the second multivariate pattern (always 0 for analysis with one multivariate pattern)
- rmin: minimal distance for GoF test
- rmax: maximal distance for GoF test
- tf: number of summary function
- was “uni” or “bi” indicates if results are from “univariate” analysis (i.e., one multivariate pattern) or “bivariate” (i.e., two multivariate patterns).
- summary function
- Delta_p_f $\Delta_f^{\mathrm{P}} = \frac{1}{S-1}\sum_{j=1}^{S}\delta_{fj}^{\mathrm{P}}$
- MPD_f $MPD_f = \sum_{i=1}\delta_{fj}^{\mathrm{P}} f_j / \sum_{i=1}\delta_{fj} f_j$
- Rank: the rank of the standard GoF test over interval rmin to rmax
- SumSt(r) the value of the summary functions at distance *r*.
- E-(r) the value of lower simulation envelop at distance *r*.
- E+(r) the value of upper simulation envelop at distance *r*.
- mean(r) the expected value of the summary functions at distance *r*.
- sig(r) indicates if there is a pointwise departure from the null model (1) or not (0)

Additionally information on the global envelope test:

- G-l lower global envelope over interval 1 to rmax/2
- G+l upper global envelope over interval 1 to rmax/2
- rank_l rank of global envelope test over interval 1 to rmax/2
- G-r lower global envelope over interval rmax/2 to radmax
- G+r upper global envelope over interval rmax/2 to radmax
- rank_r rank of global envelope test over interval rmax/2 to radmax
- G- lower global envelope over interval 1 to rmax
- G+ upper global envelope over interval 1 to rmax
- rank rank of global envelope test over interval 1 to rmax
- Effsize(r) effect size for distance r

7.4 Two multivariate patterns using a dissimilarity matrix

Programita allows you also to analyze spatial structures among individuals of two types of points of the same community. For example, you can analyze the phylogenetic diversity of small trees around large trees by using a “bivariate” phylogenetic Simpson index. In this case, the “**focal individuals**” belong to the first pattern of large trees and the “**counted individuals**” belong the second pattern of small trees. For example, the estimator of the “bivariate” phylogenetic Simpson index is formally the same as for the “univariate” phylogenetic Simpson index:

$$\beta_{phy}(r) = \sum_{f=1}^{S} \sum_{m=1}^{S} p_{fm}(r) \delta_{fm}^{P}$$

but now the mark connection functions $p_{fm}(r)$ yield the probability that, when randomly selecting a large tree and a small tree distance r apart, the large tree is of type f and the small tree of type m.

Based on the same principle, all summary functions listed in the overview table (see below) can be applied in a “bivariate” manner. The focal individuals are always taken from the first multivariate pattern (e.g., large trees) and the counted individuals are always taken from the second multivariate pattern (e.g., small trees).

Note that some of the normalization constant have also be re-interpreted accordingly. For example, in the estimation of the indices D^{P} and D_f^{P}

$$D^{\mathrm{P}} = \sum_{i=1}^{S} \sum_{j=1}^{S} \delta_{ij}^{P} f_i f_j \quad \text{and} \quad D_f^{\mathrm{P}} = \sum_{j=1}^{S} \delta_{fj}^{P} f_j \ ,$$

the f_i refers to the relative abundance of species i within the first multivariate pattern whereas the f_j refers to the relative abundance of species j within the second multivariate pattern. The same is true for of the indices S^{P}:

$$S^{\mathrm{P}} = \sum_{i=1}^{S} f_i \sum_{i=1}^{S} \delta_{ij}^{P} \ .$$

classifier				non-spatial metrics			spatial metrics			spatial
				A	B	C	A	B	C	condition
F1	α	S	community	S^{S}	S^{P}	$\Delta^{\mathrm{P}*} = S^{\mathrm{P}}/S^{\mathrm{S}}$	$\overline{ISAR}(r)$	$\overline{PISAR}(r)$	$\overline{rISAR}(r)$	$D_{fs}(r)$
F2	α	S	focal species	S_f	S_f^{P}	$\Delta_f^{\mathrm{P}} = S_f^{\mathrm{P}}/S_f$	$ISAR_f(r)$	$PISAR_f(r)$	$rISAR_f(r)$	$D_{fs}(r)$
F3	α	D	community	D	D^{P}	$c_{\mathrm{d}} = D^{\mathrm{P}}/D$	$\alpha_{\mathrm{S}}(r)$	$\alpha_{\mathrm{phy}}(r)$	$K_{\mathrm{d}}(r)$	$K_{ij}(r)/K(r)$
F4	α	D	focal species	D_f	D_f^{P}	$c_{f\mathrm{d}} = D_f^{\mathrm{P}}/D_f$	$\alpha_{f,\mathrm{S}}(r)$	$\alpha_{f,\mathrm{phy}}(r)$	$K_{f,\mathrm{d}}(r)$	$K_{ij}(r)/K(r)$
F5*	β	S	community	S^{S}	S^{P}	$\Delta^{\mathrm{P}*} = S^{\mathrm{P}}/S^{\mathrm{S}}$	$\overline{isar}(r)$	$\overline{pisar}(r)$	$\overline{risar}(r)$	$d_{fs}(r)$
F6*	β	S	focal species	S_f	S_f^{P}	$\Delta_f^{\mathrm{P}} = S_f^{\mathrm{P}}/S_f$	$isar_f(r)$	$pisar_f(r)$	$risar_f(r)$	$d_{fs}(r)$
F7	β	D	community	D	D^{P}	$c_{\mathrm{d}} = D^{\mathrm{P}}/D$	$\beta_{\mathrm{S}}(r)$	$\beta_{\mathrm{phy}}(r)$	$k_{\mathrm{d}}(r)$	$g_{ij}(r)/g(r)$
F8	β	D	focal species	D_f	D_f^{P}	$c_{f\mathrm{d}} = D_f^{\mathrm{P}}/D_f$	$\beta_{f,\mathrm{S}}(r)$	$\beta_{f,\mathrm{phy}}(r)$	$k_{f,\mathrm{d}}(r)$	$g_{ij}(r)/g(r)$

*Metric families F5 and F6 that are based on the non-cumulative probability density function $d_{ij}(r)$ of the distances to the nearest species j neighbor have not been used to date.

7.4.1 Data preparation for analysis of two multivariate pattern

For the multivariate analysis you need three data sets:

1. a data file with the location and species identity of all individuals in the two multivariate patterns. This is an ASCII file with *.phy extension.
2. a data file with the species acronyms and the species numbers. This is an ASCII file with *.txt extension
3. a data file with dissimilarity matrix. This is an ASCII file with *.txt extension

1) The data files for "bivariate" average analysis to detect phylogenetic (or functional) spatial structure in the fine-scale placement among individuals of two types are given by two multivariate patterns. For example, the focal pattern could be that of large trees and the second pattern that of small trees. The data files must be an ASCII file with *. phy extension and have the following format (the example data file DataType3bi_cluster10.phy):

```
0   316   0   316   10101
 304.98  203.69   1      1  100
 236.99  311.00   1      2  100
 289.42  171.37   1      3  100
 187.55   11.82   1      4  100
 308.96  138.19   1      5  100
   5.14  210.25   1      6  100
 176.71   76.73   2      1  100
  74.92   29.00   2      2  100
 271.00   39.00   2      3  100
 236.51  164.10   2      4  100
 227.24   31.34   2      5  100
…
```

- the first line gives the dimension of the plot (316 × 316 units) in the example and the total number of points in the list (10,101 in the example)
- the first two columns are the coordinates of the points
- the third column gives the pattern: "1" for the first focal multivariate pattern (e.g., large trees) and "2" for the second multivariate pattern (e.g., small trees)
- the forth column gives the species identifier (being an integer running from 1 to *S*).
- the fifth column is optional and can carry an quantitative mark, however, this mark is not yet used.

2) The file with species numbers and species acronyms (here file Names_random1.txt) is a **tab (or space) delimited** ASCII file with the *.txt extension and the following format:

```
1  SPECI1
2  SPECI2
…
9  SPECI9
10 SPEC10
….
```

where the first column is the species number and the second column an up to 6 letter species acronym.

3) The data file with distance matrix (here Dist_random1.txt) which is a **tab (or space) delimited** ASCII file with the *.txt extension and the following format:

```
SPECI2 SPECI1  15.9161
SPECI3 SPECI1  19.6389
SPECI3 SPECI2  19.6389
SPECI4 SPECI1  19.6389
SPECI4 SPECI2  19.6389
SPECI4 SPECI3  11.8997
SPECI5 SPECI1  19.6389
….
```

where the first two column are the six letter species acronyms of the species pair and the third column is the distance between the two species.

Note that this file must be tab or space delimited and that the species acronyms in the distance matrix and the species list must exactly match.

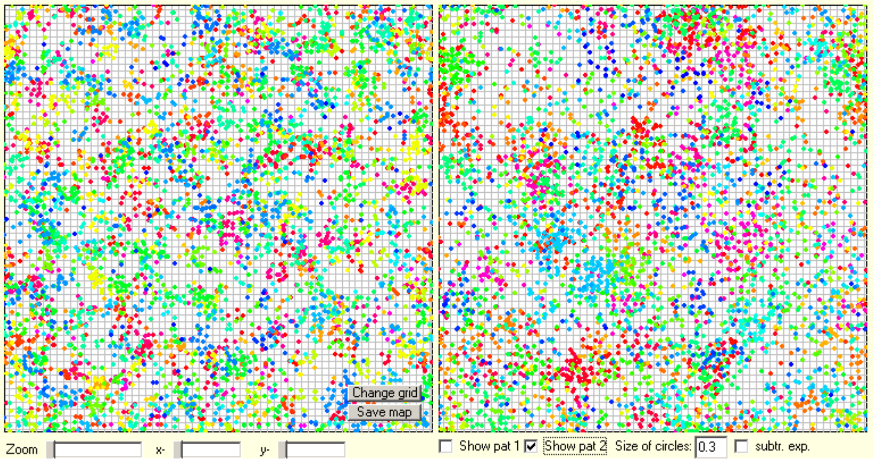
7.4.2 Two multivariate pattern and randomization of the dissimilarity matrix

The "bivariate" analysis based on randomization of the dissimilarity matrix works completely analogously to the analysis of one multivariate pattern. I provide therefore only one example that is based on the data file "DataType3bi_cluster10.phy".

The first multivariate pattern (i.e., large trees) is an example of the "dispersal limitation" communities presented in Wiegand et al. (2017). It mimicked dispersal limitation by using for each species a Thomas cluster process with parameters σ = 5m and ρ = 0.0002/m^2. Species were placed without regard to each other. The dissimilarity matrix and species abundances were taken from a "random community" generated with the R package metricTester presented in Miller et al. (2017). Thus, the spatial pattern of large trees is only governed by dispersal limitation and does not contain phylogenetic spatial structure.

The second multivariate pattern (i.e., small trees) was generated by a mixture of a random pattern (10%) and a Gaussian dispersal kernel (90%) with parameter σ = 10m where only 20% of the large trees generated offspring:

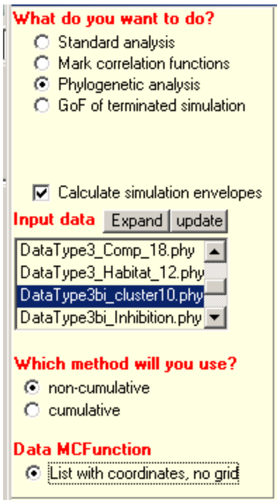
first multivariate pattern second multivariate pattern



Thus, there is also no spatial phylogenetic structure within the small trees and no spatial phylogenetic structure between small and large trees. However, due to the dispersal kernel, there is a distance decay of similarity between the community of small and large trees.

The bivariate analysis can be accessed with the following sequence of actions:

1. Select "**Phylogenetic analysis**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. Select the example file "DataType3bi_Habitat_12.phy".
3. Click "**List with coordinates, no grid**" in MCFunction
4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting "**cumulative**" in Which method will you use. However, this somewhat slows down the estimation and there the better option for estimating the cumulative index is by using the **Replicate** option.



What do you want to do?

○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

☑ Calculate simulation envelopes

Input data

DataType3_Comp_18.phy
DataType3_Habitat_12.phy
DataType3bi_cluster10.phy
DataType3bi_Inhibition.phy

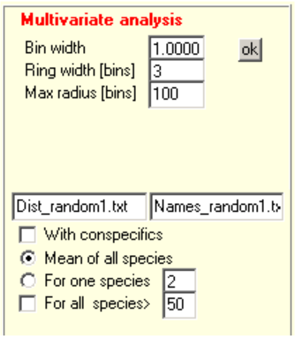
Which method will you use?

● non-cumulative
○ cumulative

Data MCFunction

● List with coordinates, no grid

5. Provide in the window Multivariate analysis the bin width in data units (here *dr* = 3), an appropriate ring width (use ring width of 1 if the analysis takes long time and then the **Replicate** option to change the ring width), and a maximal distance *r* of the analysis (here rmax = 100).
6. Provide the data file with species numbers and names (here file "Names_random1.txt")
7. Provide the data file with distance matrix (here Dist_random1.txt)
8. For the "bivariate" analysis select "**Mean of all species**"



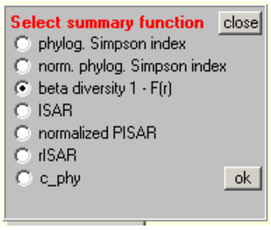
Multivariate analysis

Bin width 1.0000
Ring width [bins] 3
Max radius [bins] 100

Dist_random1.txt
Names_random1.t›

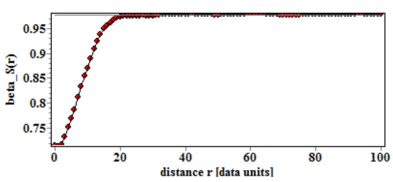
□ With conspecifics
● Mean of all species
○ For one species 2
□ For all species> 50

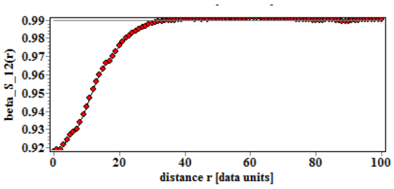
9. You can include or exclude the focal species with the check box **With conspecifics.** If you include conspecifics you obtain the phylogenetic Simpson index and if you exclude conspecifics you obtain the phylogenetic mark correlation function.
10. Click "**Calculate Index**". *Programita* now shows you the multivariate summary functions of the data. In window Select one test function you can select the different multivariate summary functions. First look at the spatially explicit Simpson index that describes species beta diversity:



Select summary function

○ phylog. Simpson index
○ norm. phylog. Simpson index
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy





The probability that a small tree at distance *r* of a large tree is heterospecific is at small distances 0.92, but increase following the Gaussian dispersal kernel to some 99% at *r* = 30m.

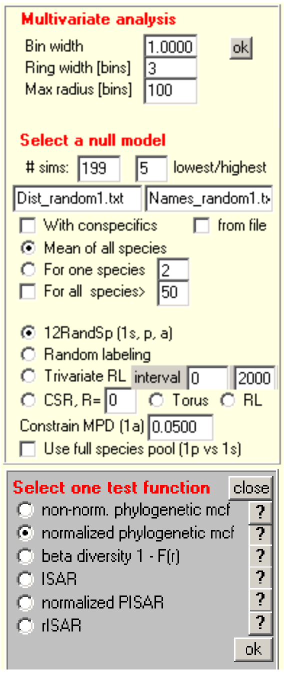
11. Click “**calculate simulation envelopes**”

12. Go to the window Multivariate analysis and select the number of simulations of the null model (199), and the rule for the simulation envelopes (5).

13. Select for the univariate analysis the species shuffling null model “**12RandomSp**” (see Shen et al. 2013). The **species shuffling null model** randomly shuffles the names of the species in the dissimilarity Matrix.

17. Disable the option “With conspecifics” ☐ With conspecifics

14. After clicking “**Calculate Index**” and running the simulations of the null model you can select in the window Select one test function among the different test functions. Note that the second multivariate pattern (i.e., small trees) were generated based on a Gaussian dispersal kernel around (20%) of the large trees. Therefore, no phylogenetic spatial structure is expected in the relationship between large and small trees.

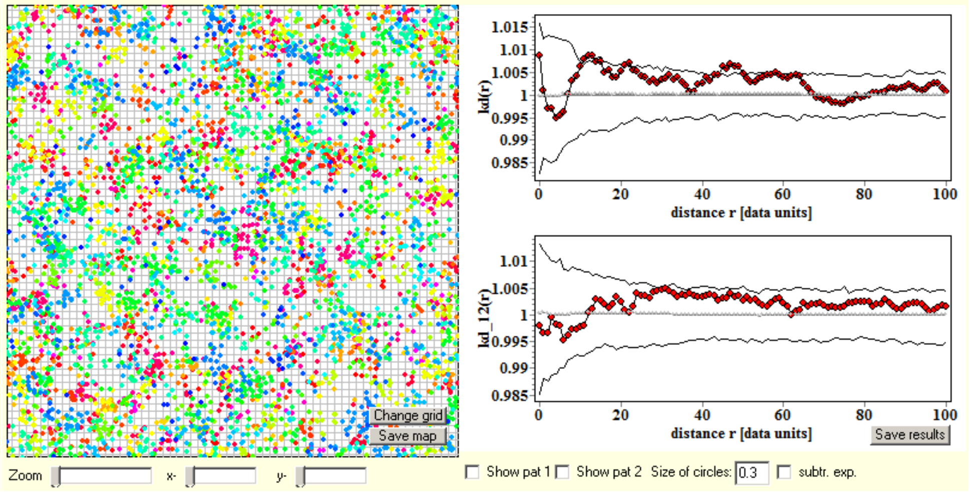


Multivariate analysis
Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 100

Select a null model
# sims: 199 5 lowest/highest
Dist_random1.txt Names_random1.tx
☐ With conspecifics ☐ from file
● Mean of all species
○ For one species 2
☐ For all species> 50
● 12RandSp (1s, p, a)
○ Random labeling
○ Trivariate RL interval 0 2000
○ CSR, R= 0 ○ Torus ○ RL
Constrain MPD (1a) 0.0500
☐ Use full species pool (1p vs 1s)

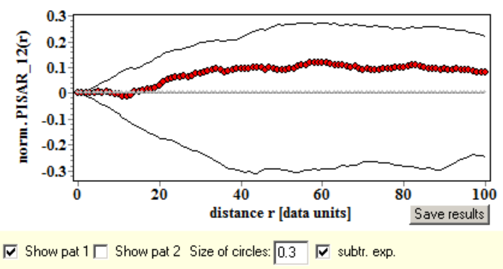
Select one test function close
○ non-norm. phylogenetic mcf ?
● normalized phylogenetic mcf ?
○ beta diversity 1 - F(r) ?
○ ISAR ?
○ normalized PISAR ?
○ rISAR ?
ok

15. The results show the pattern of large trees (multivariate pattern 1) on the left. On the right you find the uni- and bivariate summary functions. The bivariate normalized phylogenetic mark correlation function confirms absence of spatial phylogenetic structure in the “bivariate” multivariate pattern of the communities of large and small trees (right, bottom):

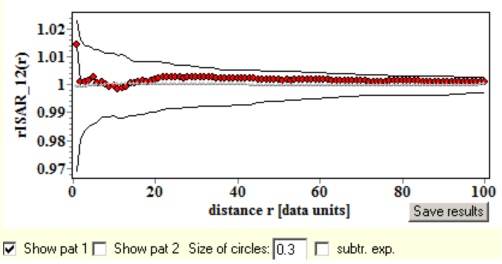


☐ Show pat 1 ☐ Show pat 2 Size of circles: 0.3 ☐ subtr. exp.

16. The same is true for the normalized *PISAR* and *rISAR* functions:



☑ Show pat 1 ☐ Show pat 2 Size of circles: 0.3 ☑ subtr. exp.



☑ Show pat 1 ☐ Show pat 2 Size of circles: 0.3 ☐ subtr. exp.

For the *PISAR* click “subtr. exp” to subtract the expectation of the null model to see the envelopes.

7.5 Multivariate analysis with a qualitative mark

Programita allows you also to analyze spatial structures based on one multivariate pattern that carries an additional qualitative mark. This mark (label) defines two types of points that are indicated by pattern 1 and pattern 2 in the same way as in the "bivariate" analysis. **The summary functions are the same as for the case of two multivariate patterns**, but the null model is different. Instead of randomizing the dissimilarity matrix, the label (qualitative mark) is randomly shuffled among the individuals of the community (i.e., the random labeling null model).

Examples for this data structure are the pattern of all surviving vs. surviving large trees in a forest community. In this case the analysis can reveal systematic differences among surviving and dead trees with respect to their phylogenetic neighborhood.

Additionally, *Programita* allows you to conduct a type of **trivariate random labeling**. Examples for this data structure are the pattern of all surviving vs. surviving small trees together with the pattern of all large trees. For example, in this case the *ISAR* function allows to find out if the species richness of dead small trees within distance *r* of large trees differs from that expected by random mortality of small trees. The phylogenetic mark correlation functions give the expected dissimilarity of large trees to surviving (or dead) small trees located at distance *r*.

7.5.1 Data preparation for random labeling analysis

For the multivariate analysis you need three data sets:

1. a data file with the location and species identity of all individuals in the two multivariate patterns. This is an ASCII file with *.phy extension.
2. a data file with the species acronyms and the species numbers. This is an ASCII file with *.txt extension
3. a data file with dissimilarity matrix. This is an ASCII file with *.txt extension

1) The data files for analysis with a qualitative mark is given by one multivariate pattern which comprises two types of points (e.g., surviving vs. dead) that represent a qualitative mark. The data files must be an ASCII file with *. phy extension and have the following format (the example data file DataType3_RL_Habitat_12.phy):

```
0   300  0   300   12000
0.02   17.71   1   5
0.02  137.32   2  10
0.19    2.96   2  10
0.22  214.77   2   2
…
```

- the first line gives the dimension of the plot (300 × 300 units) in the example and the total number of points in the list (12,000 in the example)
- the first two columns are the coordinates of the points
- the third column gives the qualitative mark and must be "1" or "2"
- the forth column gives the species identifier (being an integer running from 1 to *S*).
- the fifth column is optional and can carry an quantitative mark, however, this mark is not yet used.

2) The file with species numbers and species acronyms (here file sim_12.txt) is a **tab or space delimited** ASCII file with the *.txt extension and the following format:

```
1   SPECI1
2   SPECI2
3   SPECI3
…
10  SPEC10
```

where the first column is the species number and the second column a 6 letter species acronym.

3) The data file with distance matrix (here DistanceMatrix_Habitat_12.txt) which is a **tab or space delimited** ASCII file with the *.txt extension and the following format:

```
SPECI1  SPECI2  0.7587
SPECI1  SPECI3  0.9752
SPECI1  SPECI4  0.6250
SPECI1  SPECI5  0.4814
SPECI1  SPECI6  0.0504
SPECI1  SPECI7  0.1111
SPECI1  SPECI8  0.4236
SPECI1  SPECI9  0.0211
SPECI1  SPEC10  0.5394
```

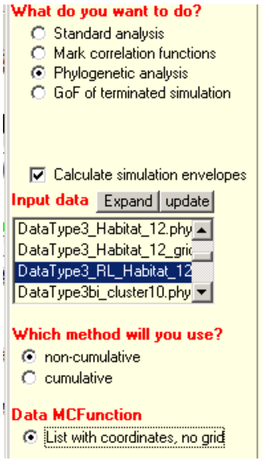
where the first two column are the six letter species acronyms of the species pair and the third column is the distance between the two species.

Note that this file must be tab or space delimited and that the species acronyms in the distance matrix and the species list must exactly match.

7.5.2 Multivariate analysis with a qualitative mark

The analysis of a **multivariate pattern with qualitative mark** works completely analogously to the bivariate average analysis. The example is based on the data file "DataType3RL_Habitat_12.phy" which is identical to the data file for the univariate average analysis (DataType3_Habitat_12.phy), except that the label "1" or "2" was randomly assigned to the individuals of the community to yield approximately the same number of points of type 1 and type 2. This analysis can be accessed with the following sequence of actions:

1. Select "**Phylogenetic analysis**" in What do you want to do?
2. Highlight data file you want to analyze in Input data. Select the example file "DataType3_RL_Habitat_12.phy".
3. Click "**List with coordinates, no grid**" in MCFunction
4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting "**cumulative**" in Which method will you use. However, this somewhat slows down the estimation and there the better option for estimating the cumulative index is by using the **Replicate** option.



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

☑ Calculate simulation envelopes

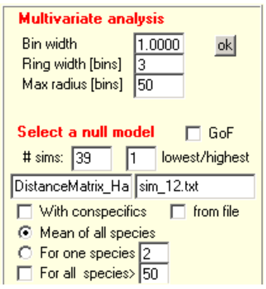
Input data

DataType3_Habitat_12.phy
DataType3_Habitat_12_gric
DataType3_RL_Habitat_12
DataType3bi_cluster10.phy

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
● List with coordinates, no grid

5. Provide in the window Multivariate analysis the bin width in data units, an appropriate ring width (use ring width of 1 if the analysis takes long time and then the **Replicate** option to change the ring width), and a maximal distance *r* of the analysis.



Multivariate analysis
Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50
Select a null model □ GoF
# sims: 39 1 lowest/highest
DistanceMatrix_Ha sim_12.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

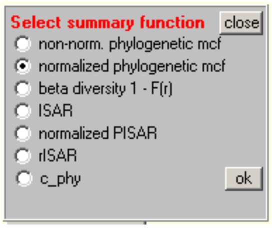
6. Provide the data file with species numbers and names (here file "sim_12.txt")

7. Provide the data file with distance matrix (here DistanceMatrix_Habitat_12.txt)

8. For the community level analysis select "**Mean of all species**". The species level analysis (i.e., "For one species") does not really make sense here.

9. You can include or exclude the focal species with the check box **With conspecifics.** If you include conspecifics you obtain the phylogenetic Simpson index and if you exclude conspecifics you obtain the phylogenetic mark correlation function.

10. Click "**Calculate Index**". *Programita* now shows you the multivariate summary functions of the data. In window Select one test function you can select the different multivariate summary functions.



Select summary function close
○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy ok

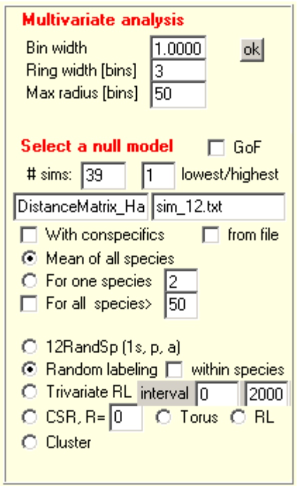
11. Click "**calculate simulation envelopes**"

12. Go to the window Multivariate analysis and select the number of simulations of the null model (39), and the rule for the simulation envelopes (1). For demonstrative purpose use 39 simulations, but for serious analysis use at least 199.

13. Select for the univariate analysis the random labeling null model "**RandomLabeling**". This null model randomly shuffles the labels "1" and "2" of the individuals of the community. It is much slower than the species shuffling null model.

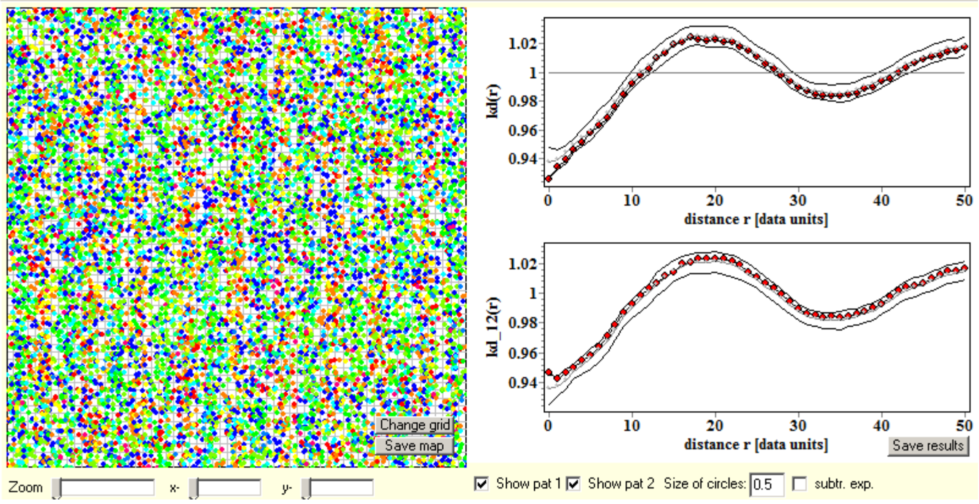
18. Disable the option "With conspecifics" □ With conspecifics

14. After clicking "**Calculate Index**" and running the simulations of the null model you can select in the window Select one test function among the different test functions.



Multivariate analysis
Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50
Select a null model □ GoF
# sims: 39 1 lowest/highest
DistanceMatrix_Ha sim_12.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50
○ 12RandSp (1s, p, a)
● Random labeling □ within species
○ Trivariate RL interval 0 2000
○ CSR, R= 0 ○ Torus ○ RL
○ Cluster

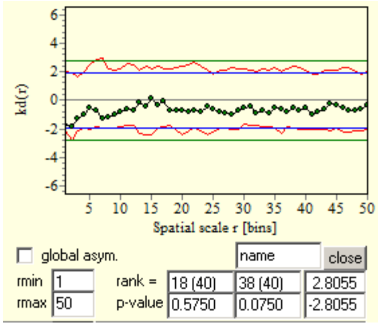
15. With “Show pat 1” enabled, *Programita* shows the type 1 points of the qualitatively marked multivariate pattern and with “Show pat 2” enabled, *Programita* shows the type 2 points. On the right you find the uni- and bivariate summary functions:



☑ Show pat 1 ☑ Show pat 2 Size of circles: 0.5 □ subtr. exp.

The analysis shows clearly that the phylogenetic neighborhood does not differ between points of type 1 and type 2. This was expected because the label was randomly assigned to the points. This finding is also supported by the GoF test with the student transformation:

type 1 around type 1



□ global asym.

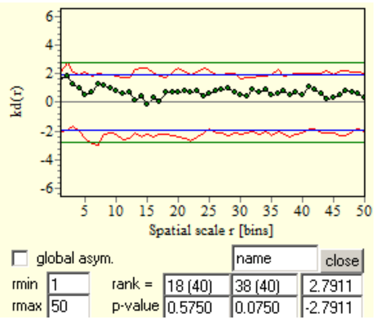
rmin 1

rmax 50

rank = 18 (40) 38 (40) 2.8055

p-value 0.5750 0.0750 -2.8055

type 2 around type 1



□ global asym.

rmin 1

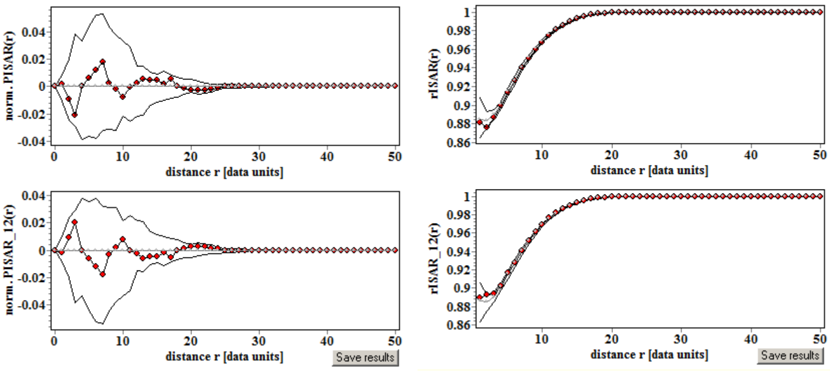
rmax 50

rank = 18 (40) 38 (40) 2.7911

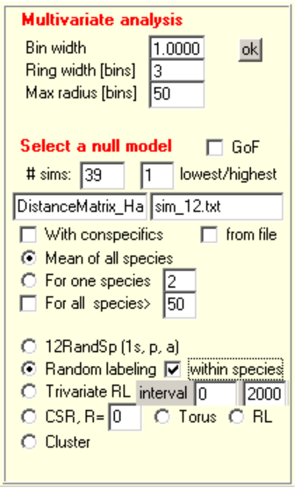
p-value 0.5750 0.0750 -2.7911

16. Note that when using the random labeling null model the dissimilarity matrix is the same in all simulations of the null model.

17. As expected, the *PISAR* and *rISAR* functions do also not show any departures from the random labeling null model:



Random labeling within species



Multivariate analysis

Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50

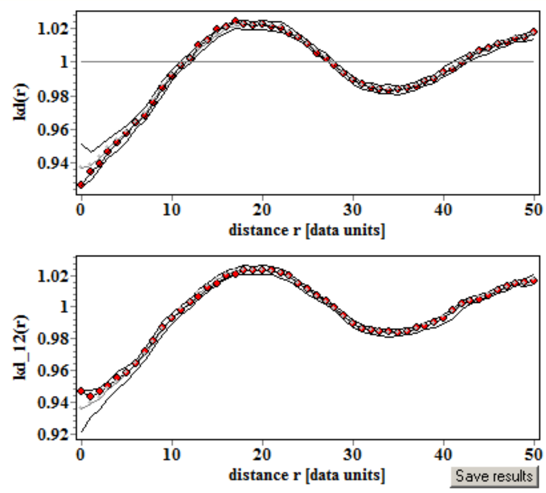
Select a null model □ GoF
# sims: 39 1 lowest/highest
DistanceMatrix_Ha sim_12.txt
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

○ 12RandSp (1s, p, a)
● Random labeling ☑ within species
○ Trivariate RL interval 0 2000
○ CSR, R= 0 ○ Torus ○ RL
○ Cluster

Similar to the standard random labeling for a simple qualitatively marked pattern, *Programita* allows you also to conduct the random labeling inside of the different species. If the label is surviving vs. dead, this null model thus conserves the observed mortality rates within species.

To conduct random labeling inside of the different species enable the checkbox “**within species**” on the right of the Random labeling option.

The results of the example show that the pointwise simulation envelopes become narrower which was expected because the variability in the assignment of type 1 or type 2 becomes smaller when forcing each species to maintain the frequency of type 1 and type 2 points.



7.5.3 Multivariate trivariate random labeling

Programita allows you also to analyze spatial structures among individuals of two multivariate patterns were one of them is a qualitatively marked pattern. In this case you can study the potential impact of a focal multivariate pattern (e.g., large trees) on the qualitative marking of a second multivariate pattern (e.g., surviving vs. dead small trees). Thus, in this data structure you have:

- the focal multivariate pattern (codes as type 3, e.g., large trees)
- the second qualitatively marked multivariate pattern with
    - the pattern of type 1 (e.g., dead small trees)
    - the pattern of type 2 (e.g., surviving small trees)

The analysis allows to find out if the species, phylogenetic or functional diversity of (say) dead small trees in the neighborhood of large trees differs from that of all (surviving and dead) small trees in the neighborhood of large trees. In contrast to “standard” trivariate random labeling that works at the species level (i.e., you have one univariate focal pattern and a qualitatively marked pattern), the multivariate trivariate random labeling works at the community level and is able to quantify additionally the impact of phylogenetic or functional distances among species.

Using the *ISAR* family of summary functions, you can assess if the species richness (or the phylogenetic diversity) of dead trees within distance *r* of large trees differs from that of all small trees (i.e., surviving and dead).

Using the Simpson family of summary functions, you can assess for example with the $\beta_{\text{phy}}(r)$ if the expected dissimilarity between a randomly selected large tree and a dead small tree at distance *r* differs from that of all small trees. If it would be significantly smaller, this means that small trees of species more similar to the focal large trees tend to have at distance *r* a higher risk of mortality than more distantly related small trees.

7.5.4 Data preparation for trivariate random labeling analysis

For the multivariate analysis of trivariate random labeling you need three data sets:

1. a data file with the location and species identity of all individuals in the two multivariate patterns. This is an ASCII file with *.phy extension.
2. a data file with the species acronyms and the species numbers. This is an ASCII file with *.txt extension
3. a data file with dissimilarity matrix. This is an ASCII file with *.txt extension

1) The data files for analysis with a qualitative mark is given by one multivariate pattern which comprises two types of points (e.g., surviving vs. dead) that represent a qualitative mark. The data files must be an ASCII file with *. phy extension and have the following format (the example data file DataType3triRL_cluster10.phy):

```
0 316 0 316 10101
  304.98  203.69  3  1  100
  305.31  219.24  3  1  100
  289.91  249.78  3  1  100
  176.71   76.73  1  1  100
   34.70  199.96  2  1  100
  155.62  185.90  1  1  100
  159.68  168.80  1  1  100…
```

- the first line gives the dimension of the plot (316 × 316 units) in the example and the total number of points in the list (10,101 in the example)
- the first two columns are the coordinates of the points
- the third column gives the qualitative mark and must be “1” or “2” for the multivariate pattern with the qualitative mark (e.g., surviving or dead small trees) and “3” for the focal pattern (e.g., large trees)
- the forth column gives the species identifier (being an integer running from 1 to *S*).
- the fifth column is optional and can carry an quantitative mark, however, this mark is not yet used.

2) The file with species numbers and species acronyms (here Names_random1.txt) is a **tab or space delimited** ASCII file with the *.txt extension and the following format:

```
1   SPECI1
2   SPECI2
3   SPECI3
4   SPECI4
…
```

where the first column is the species number and the second column a 6 letter species acronym.

3) The data file with distance matrix (here Dist_Random1.txt) which is a **tab or space delimited** ASCII file with the *.txt extension and the following format:

```
SPECI2 SPECI1  15.9161
SPECI3 SPECI1  19.6389
SPECI3 SPECI2  19.6389
SPECI4 SPECI1  19.6389
SPECI4 SPECI2  19.6389
SPECI4 SPECI3  11.8997
```

where the first two column are the six letter species acronyms of the species pair and the third column is the distance between the two species.

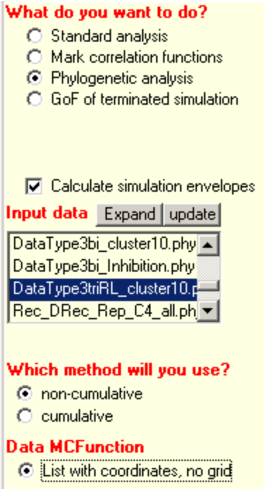
Note that this file must be tab or space delimited and that the species acronyms in the distance matrix and the species list must exactly match.

7.5.5 Example of multivariate trivariate random labeling

The example data set is based on “DataType3bi_cluster10.phy”, where the multivariate pattern of large trees are an example of the “dispersal limitation” communities presented in Wiegand et al. (2017) and the small trees were generated by a mixture of a random pattern (10%) and a Gaussian dispersal kernel (90%) around large trees with parameter $\sigma = 10$m where only 20% of the large trees generated offspring. The qualitative mark of the pattern of small trees was randomly assigned.

In the example data file “DataType3triRL_cluster10.phy”, the focal pattern of large trees was coded with value “3”, and the qualitatively marked pattern of surviving and dead small trees was coded with value “1” (say dead) and “2” (say surviving”). This analysis can be accessed with the following sequence of actions:

1. Select “**Phylogenetic analysis**” in What do you want to do?
2. Highlight data file you want to analyze in Input data. Select the example file “DataType3triRL_cluster10.phy”.
3. Click “**List with coordinates, no grid**” in MCFunction
4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting “**cumulative**” in Which method will you use. However, this somewhat slows down the estimation and there the better option for estimating the cumulative index is by using the **Replicate** option.



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

☑ Calculate simulation envelopes

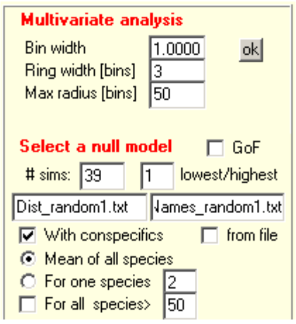
Input data Expand update

DataType3bi_cluster10.phy
DataType3bi_Inhibition.phy
DataType3triRL_cluster10.p
Rec_DRec_Rep_C4_all.ph

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
● List with coordinates, no grid

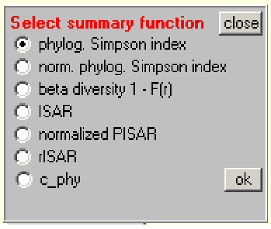
5. Provide in the window Multivariate analysis the bin width in data units, an appropriate ring width (use ring width of 1 if the analysis takes long time and then the **Replicate** option to change the ring width), and a maximal distance *r* of the analysis.
6. Provide the data file with species numbers and names (here file “Names_random1.txt”)
7. Provide the data file with distance matrix (here Dist_random1.txt)
8. For the community level analysis select “**Mean of all species**”



Multivariate analysis
Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50

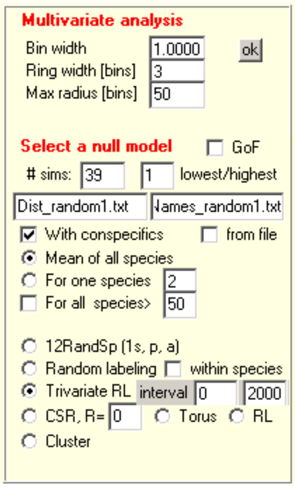
Select a null model □ GoF
# sims: 39 1 lowest/highest
Dist_random1.txt Names_random1.txt
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

9. You can include or exclude the focal species with the check box **With conspecifics.** If you include conspecifics you obtain the phylogenetic Simpson index and if you exclude conspecifics you obtain the phylogenetic mark correlation function.
10. Click “**Calculate Index**”. *Programita* now shows you the multivariate summary functions of the data. In window Select one test function you can select the different multivariate summary functions.



Select summary function close
● phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy ok

11. Click “**calculate simulation envelopes**”
12. Go to the window Multivariate analysis and select the number of simulations of the null model (39), and the rule for the simulation envelopes (1). For demonstrative purpose use 39 simulations, but for serious analysis use at least 199.
13. Select for the univariate analysis the random labeling null model “**TrivariateRL**”. This null model randomly shuffles the labels “1” and “2” of the individuals of the community. It is much slower than the species shuffling null model.
14. After clicking “**Calculate Index**” and running the simulations of the null model you can select in the window Select one test function among the different test functions.

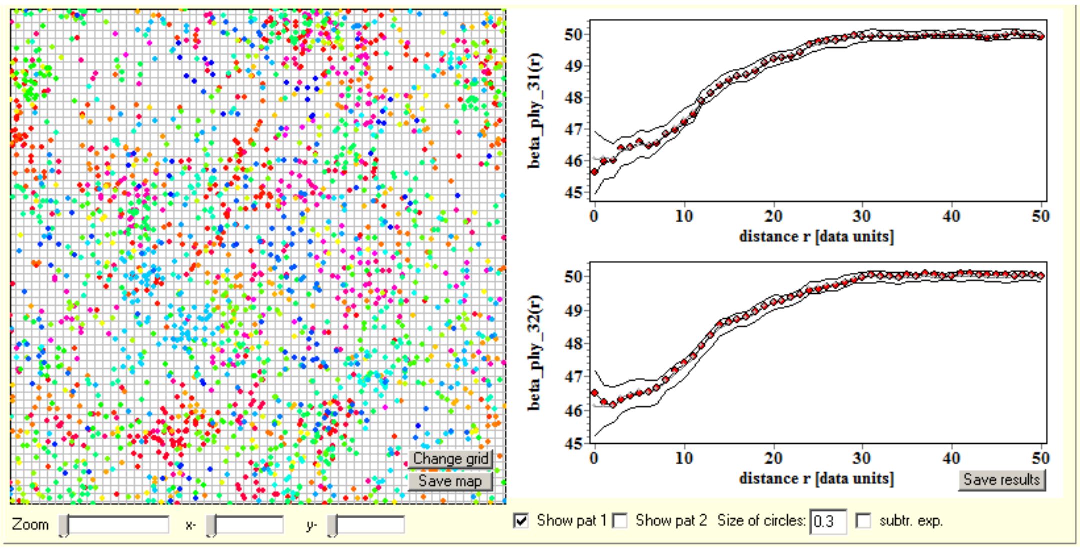


Multivariate analysis
Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50

Select a null model □ GoF
# sims: 39 1 lowest/highest
Dist_random1.txt Names_random1.txt
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

○ 12RandSp (1s, p, a)
○ Random labeling □ within species
● Trivariate RL interval 0 2000
○ CSR, R= 0 ○ Torus ○ RL
○ Cluster

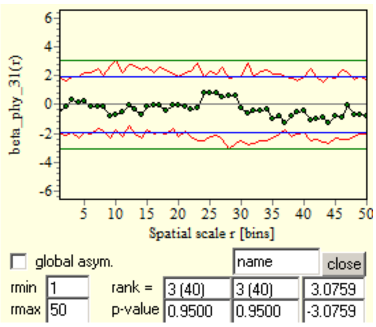
15. The results show the pattern on the left. With “Show pat 1” enabled, *Programita* shows the type 1 points of the qualitatively marked multivariate pattern and with “Show pat 2” enabled, *Programita* shows the type 2 points of the qualitatively marked multivariate pattern. On the right you find trivariate summary functions where the upper panels shows the relationship between the focal (type 3) points and type 1 points and the lower panels shows the relationship between the focal (type 3) points and type 2 points:



☑ Show pat 1 □ Show pat 2 Size of circles: 0.3 □ subtr. exp.

As expected, the analysis with the phylogenetic Simpson index $\beta_{\text{phy}}(r)$ shows that the expected dissimilarity between a randomly selected large tree (type “3”) and a surviving small tree (type “1”) located at distance r does not differ from that of the random mortality null model (top figure). The lower figure shows the analogous summary function for dead small trees (type “2”) around large trees. Lack of spatial phylogenetic structure in the relationship between small dead trees and large trees is also supported by the GoF test with the student transformation:

type 1 around type 1

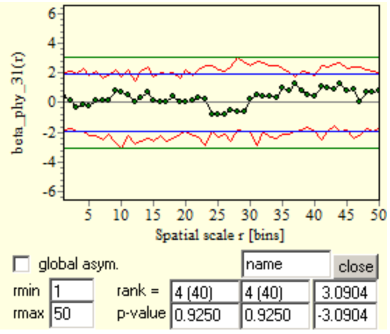


□ global asym.

rmin 1
rmax 50

rank = 3 (40) 3 (40) 3.0759
p-value 0.9500 0.9500 -3.0759

type 2 around type 1



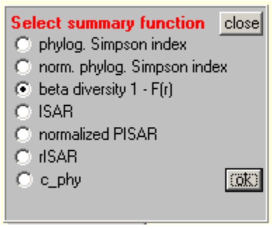
□ global asym.

rmin 1
rmax 50

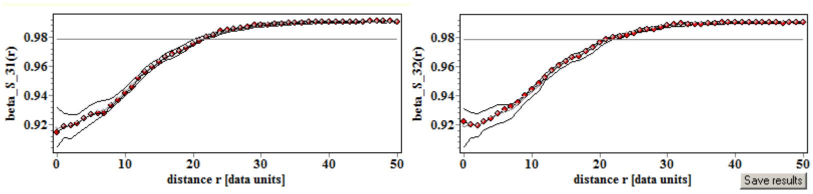
rank = 4 (40) 4 (40) 3.0904
p-value 0.9250 0.9250 -3.0904

16. Note that when using the random labeling null model the dissimilarity matrix is the same in all simulations of the null model.

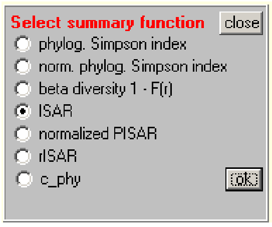
17. The species beta diversity captured by the Simpson index $\beta_S(r)$ does also not show any departures from the null model:



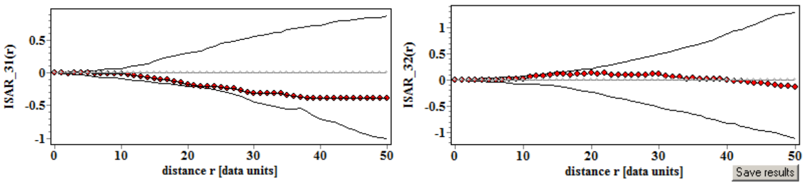
○ phylog. Simpson index
○ norm. phylog. Simpson index
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy



18. The same is true for the *ISAR*:

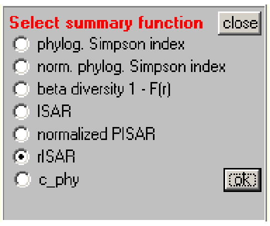


○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy

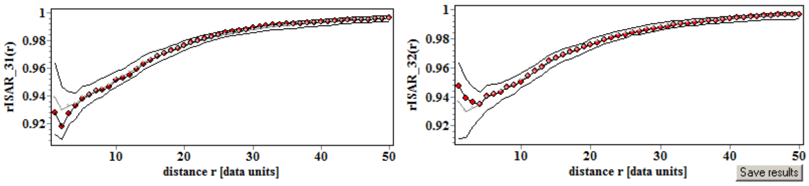


indicating that the species richness of dead small trees around large trees does not differ from the null expectation.

19. As expected, the *rISAR* shows that the expected pairwise dissimilarity between the typical large tree and dead small trees of all other species within distance *r* do not differ from the null expectation:



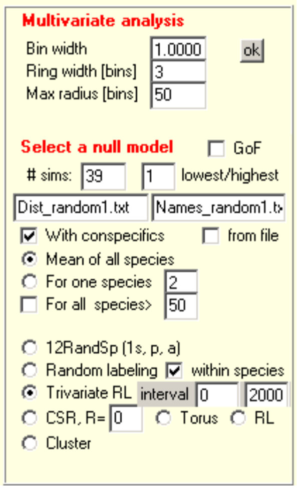
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
● rISAR
○ c_phy



Random labeling within species

Similar to the standard random labeling for a simple qualitatively marked pattern, *Programita* allows you also to conduct the random labeling inside of the different species. If the label is surviving vs. dead, this null model thus conserves the observed mortality rates within species.

To conduct random labeling inside of the different species enable the checkbox "**within species**" on the right of the Random labeling option.



Bin width 1.0000
Ring width [bins] 3
Max radius [bins] 50

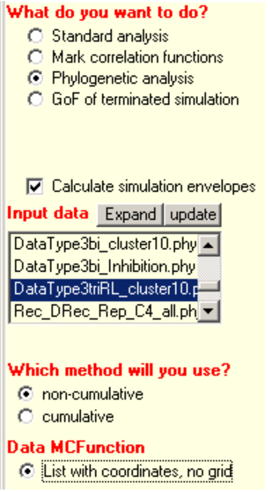
□ GoF
# sims: 39 1 lowest/highest
Dist_random1.txt Names_random1.tx
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50
○ 12RandSp (1s, p, a)
○ Random labeling ☑ within species
● Trivariate RL interval 0 2000
○ CSR, R= 0 ○ Torus ○ RL
○ Cluster

7.5.6 Individual analysis of trivariate random labeling

Programita allows you to conduct individual analyses of trivariate random labeling based on the focal pattern of a given species (extracted from the multivariate focal pattern of type "3") and a second, qualitatively marked multivariate pattern.

For example, in the simplest case you can explore using the *ISAR* function if the species richness of dead small trees around the large trees of the given focal species *f* differs from that expected under random mortality of the small trees. When considering additionally phylogenetic or functional dissimilarities between species, you can explore using the *PISAR* function if large trees of a given species impact small trees in their neighborhood in a way that the phylogenetic (or functional diversity) of dead small trees differs from that expected under random mortality. One expectation would be that small trees of species more similar to the large focal individuals die with higher probability than more dissimilar species.

The **individual multivariate analyses of trivariate random labeling** can be accessed with the same sequence of actions as the standard average analysis:



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

☑ Calculate simulation envelopes

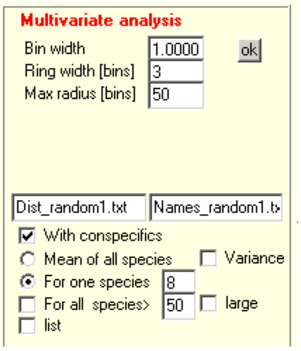
Input data

DataType3bi_cluster10.phy
DataType3bi_Inhibition.phy
DataType3triRL_cluster10.p
Rec_DRec_Rep_C4_all.ph

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
● List with coordinates, no grid

1. Select "**Phylogenetic analysis**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data. Select the example file "DataType3tri_cluster10.phy".
3. Click "**List with coordinates, no grid**" in MCFunction
4. Provide in the window Multivariate analysis the bin width in data units (1), an appropriate ring width (use ring width of 1 if the analysis takes long time and then the **Replicate** option to change the ring width), and a maximal distance *r* of the analysis (50).

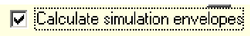


Multivariate analysis
Bin width 1.0000
Ring width [bins] 3
Max radius [bins] 50

Dist_random1.txt Names_random1.t
☑ With conspecifics
○ Mean of all species ☐ Variance
● For one species 8
☐ For all species> 50 ☐ large
☐ list

5. Provide the data file with species numbers and names (here file "Names_random1.txt")
6. Provide the data file with distance matrix (here Dist_random1.txt)
7. For the individual analysis select "**One focal species**" and select the species number of the focal species (8 in the example)
8. You can include or exclude the focal species from the count pattern *m* with the check box **With conspecifics**. If you include conspecifics you obtain the individual phylogenetic Simpson index and if you exclude conspecifics you obtain the individual phylogenetic mark correlation function (see section 3.1.7.6 in Wiegand and Moloney 2014 and Shen et al. 2013).

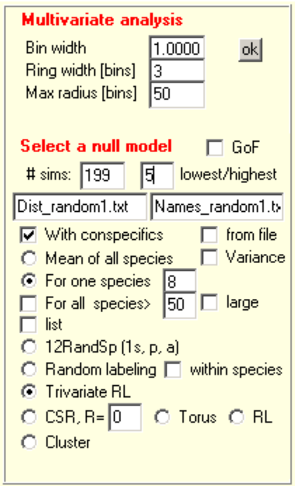
9. Click “**calculate simulation envelopes**”



☑ Calculate simulation envelopes

10. Go back to the window Multivariate analysis and select the number of simulations of the null model (199), and the rule for the simulation envelopes (5).

11. Select for the univariate analysis the random labeling null model “**TrivariateRL**”. This null model randomly shuffles the labels “1” and “2” of the individuals of the community. It is much slower than the species shuffling null model.



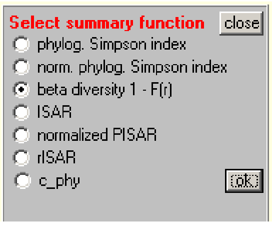
Multivariate analysis

Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50

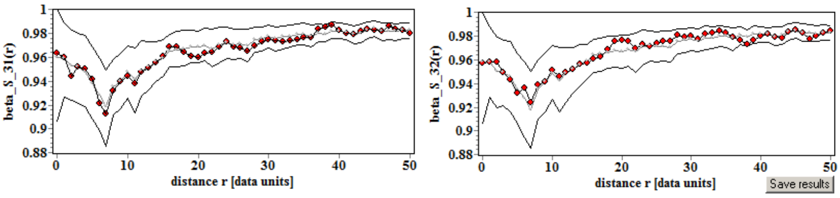
Select a null model □ GoF
# sims: 199 5 lowest/highest
Dist_random1.txt Names_random1.t
☑ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 8
□ For all species> 50 □ large
□ list
○ 12RandSp (1s, p, a)
○ Random labeling □ within species
● Trivariate RL
○ CSR, R= 0 ○ Torus ○ RL
○ Cluster

12. After clicking “**Calculate Index**” and running the simulations of the null model you can select in the window Select one test function among the different test functions.

13. Select first “beta diversity 1 - F(r)” to obtain the individual spatially-explicit Simpson index that yields the probability that a dead small tree distance *r* away from a large tree of the focal species 8 is a heterospecific:

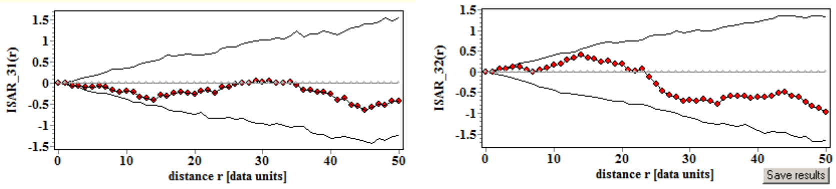


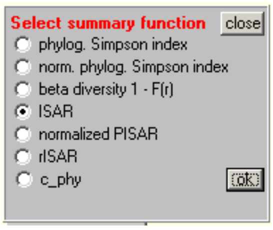
Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy ok



The analysis shows that there is, as expected by construction of the data set, no difference to the random mortality null model.

14. A similar result holds for the *ISAR* function:

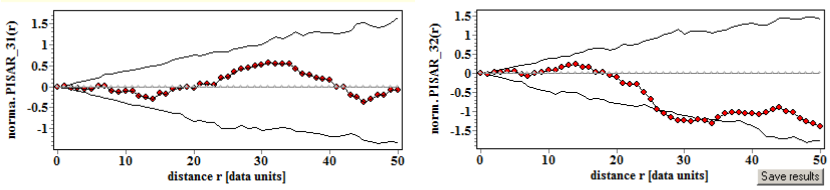


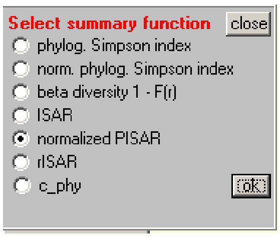


Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy ok

The species richness of small trees around large trees of the focal species 8 do nut differ between surviving and dead small trees.

15. The PISAR functions do not indicate differences in the phylogenetic diversity of surviving vs. dead small trees in the neighborhood of large focal trees:



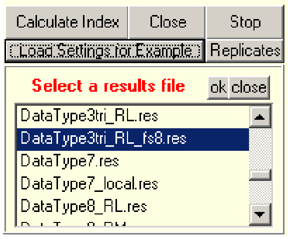


Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
● normalized PISAR
○ rISAR
○ c_phy ok

7.5.7 Series of individual analysis of multivariate trivariate random labeling

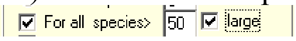
When running individual analyses one is in general interested to obtain results for all species in the community to understand variability in individual responses. Therefore, Programita offers the convenient possibility to run individual analyses of all focal species that have a minimum number of individuals. *Programita* outputs for this series analysis an additional file with a results summary. To run the series of analyses based on the settings of the previous example follow the following steps:

1. To load the settings from the previous example use the “**Load Settings for Example**” option. Highlight the file DataType3tri_RL_fs8.res and click the small ok button. *Programita* now loads all settings from this analysis.



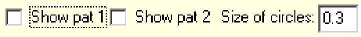
DataType3tri_RL.res
DataType3tri_RL_fs8.res
DataType7.res
DataType7_local.res
DataType8_RL.res

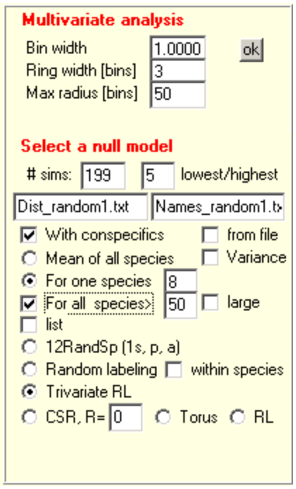
2. Enable the checkbox “**Run all focal species**” and provide the minimal number of individuals of the focal species (default value is 50). To save the *.rep file, disable the checkbox “large”:


☑ For all species> 50 ☑ large

3. Select a summary function in window Select one test function, the *.res results file will use this test function.

4. Click “**Calculate Index**” and *Programita* runs the individual analyses of all focal species with more than 50 individuals. To speed up to estimation, disable the options that plot the focal pattern after each simulation of the null model:

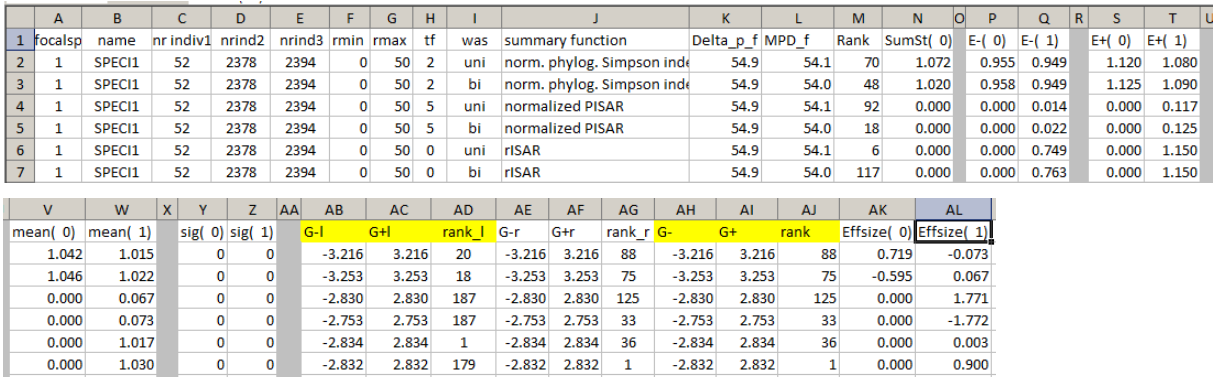

□ Show pat 1 □ Show pat 2 Size of circles: 0.3



Bin width 1.0000
Ring width [bins] 3
Max radius [bins] 50
# sims: 199 5 lowest/highest
Dist_random1.txt Names_random1.tx
☑ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 8
☑ For all species> 50 □ large
□ list
○ 12RandSp (1s, p, a)
○ Random labeling □ within species
● Trivariate RL
○ CSR, R= 0 ○ Torus ○ RL

5. *Programita* generates results files “name_fsp_nr_phy.res” and “mcf_name_fsp_nr_phy.rep” where “name” is the file name (here “DataType3triRL_cluster10”) and “nr” the number of the focal species. For example, the first results file in the series for species 1 is DataType3triRL_cluster10_fsp_1_phy.res.

6. *Programita* also outputs a file with a results summary named “name.txt” where the name is the name of the data file (here “DataType3_Habitat_12”):



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	focalsp	name	nr indiv1	nrind2	nrind3	rmin	rmax	tf	was	summary function	Delta_p_f	MPD_f	Rank	SumSt(0)		E-(0)	E-(1)		E+(0)	E+(1)
2	1	SPECI1	52	2378	2394	0	50	2	uni	norm. phylog. Simpson inde	54.9	54.1	70	1.072		0.955	0.949		1.120	1.080
3	1	SPECI1	52	2378	2394	0	50	2	bi	norm. phylog. Simpson inde	54.9	54.0	48	1.020		0.958	0.949		1.125	1.090
4	1	SPECI1	52	2378	2394	0	50	5	uni	normalized PISAR	54.9	54.1	92	0.000		0.000	0.014		0.000	0.117
5	1	SPECI1	52	2378	2394	0	50	5	bi	normalized PISAR	54.9	54.0	18	0.000		0.000	0.022		0.000	0.125
6	1	SPECI1	52	2378	2394	0	50	0	uni	rISAR	54.9	54.1	6	0.000		0.000	0.749		0.000	1.150
7	1	SPECI1	52	2378	2394	0	50	0	bi	rISAR	54.9	54.0	117	0.000		0.000	0.763		0.000	1.150

V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
mean(0)	mean(1)		sig(0)	sig(1)		G-l	G+l	rank_l	G-r	G+r	rank_r	G-	G+	rank	Effsize(0)	Effsize(1)
1.042	1.015		0	0		-3.216	3.216	20	-3.216	3.216	88	-3.216	3.216	88	0.719	-0.073
1.046	1.022		0	0		-3.253	3.253	18	-3.253	3.253	75	-3.253	3.253	75	-0.595	0.067
0.000	0.067		0	0		-2.830	2.830	187	-2.830	2.830	125	-2.830	2.830	125	0.000	1.771
0.000	0.073		0	0		-2.753	2.753	187	-2.753	2.753	33	-2.753	2.753	33	0.000	-1.772
0.000	1.017		0	0		-2.834	2.834	1	-2.834	2.834	36	-2.834	2.834	36	0.000	0.003
0.000	1.030		0	0		-2.832	2.832	179	-2.832	2.832	1	-2.832	2.832	1	0.000	0.900

10. The results summary provides information on:

- focalsp: the number of the focal species
- name: the acronym of the focal species
- nrindiv1: number of individuals of type 1 of the qualitatively marked pattern
- nrind2: number of individuals of type 2 of the qualitatively marked pattern
- nrind3: number of individuals of the focal multivariate pattern
- rmin: minimal distance for GoF test
- rmax: maximal distance for GoF test
- tf: number of summary function
- was “uni” or “bi” indicates if results are shown for summary function that quantify pairs of type 3 -1 (uni) and 3 - 2 (bi) points.
- summary function: the summary function
- Delta_p_f $\Delta^{\mathrm{P}}_{f} = \frac{1}{S-1}\sum_{j=1}^{S}\delta^{\mathrm{P}}_{fj}$
- MPD_f $MPD_f = \sum_{i=1}\delta^{\mathrm{P}}_{fj} f_j / \sum_{i=1}\delta_{fj} f_j$ where *f* is the focal species and *j* indicates type 1 points (univariate) and type 2 points (bivariate).
- Rank: the rank of the standard GoF test over interval rmin to rmax
- SumSt(r) the value of the summary functions at distance *r*.
- E-(r) the value of lower simulation envelop at distance *r*.
- E+(r) the value of upper simulation envelop at distance *r*.
- mean(r) the expected value of the summary functions at distance *r*.
- sig(r) indicates if there is a pointwise departure from the null model (1) or not (0)

Additionally information on the global envelope test:

- G-l lower global envelope over interval 1 to rmax/2
- G+l upper global envelope over interval 1 to rmax/2
- rank_l rank of global envelope test over interval 1 to rmax/2
- G-r lower global envelope over interval rmax/2 to radmax
- G+r upper global envelope over interval rmax/2 to radmax
- rank_r rank of global envelope test over interval rmax/2 to radmax
- G- lower global envelope over interval 1 to rmax
- G+ upper global envelope over interval 1 to rmax
- rank rank of global envelope test over interval 1 to rmax
- Effsize(r) effect size for distance r

7.6 Combine replicates for multivariate analysis

7.6.1 Combine replicates

In some cases you may have data of several replicate plots of a larger point pattern under identical conditions. In this case the resulting summary functions of the individual replicate plots can be combined into average summary functions. This is of particular interest if the number of points in each replicate plot is relatively low. In this case the simulation envelopes of individual analyses would become wide, but combining the data of several replicate plots into average summary functions increases the sample size and thus narrows the simulation envelopes.

Note that combining replicates works in the multivariate case only if the species list and the dissimilarity matrix is the same for all replicate plots (i.e., it must be a common matrix and a common species list including all species in all plots), but there is no problem if some species occur only in some of the plots.

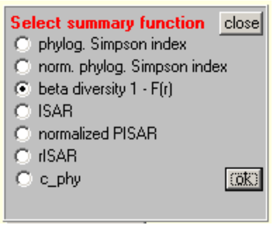
Combine replicates for multivariate analysis works in the same way as for mark correlation functions. After each multivariate analysis with a distance matrix a temporary file MCF_test.dat is created which is then renamed into mcf_name_phy.rep after saving results.

This is an example for the first few lines columns of the univariate part of a mcf_name_phy.rep output file:

```
simnr     r   MCF11_t0  MCF11_t1     MCF11_t2  MCF11_t3    MCF11_t4 MCF11_t5     Zaehler11
    0     0   0.994660 611.599757   0.985143   0.348398   0.348398 212.997979     3286.00
    0     1   0.997803 614.284335   0.989467   1.158284   1.158284 710.371146     8286.00
    0     2   0.997525 610.901445   0.984018   2.273656   2.273656 1394.035937   12460.00
    0     3   0.999353 613.801751   0.988690   3.564693   3.564693 2189.610864   15758.00
    0     4   1.001270 615.290318   0.991088   4.963715   4.963715 3054.806301   18902.00
    0     5   1.001618 613.680947   0.988495   6.460728   6.460728 3977.493102   21960.00
    0     6   1.003077 617.437710   0.994547   8.035959   8.035959 4954.475959   24598.00
    0     7   1.003720 617.016484   0.993868   9.637697   9.637697 5945.819098   27250.00
```

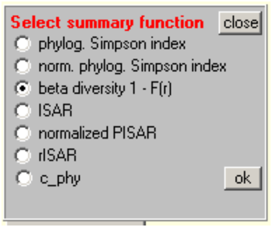
The columns of the file contain the following information:

- simnr: number of simulation of the null model where 0 indicates the observed data and 1, 2, … are the simulations of the null model.
- r: the distance bin
- MCF11_t0, MCF11_t1, …, MCF11_t5, MCF11_t6: the values of the different "univariate" summary functions where
    - t1 refers to the non-normalized phylogenetic mark correlation function or the phylogenetic Simpson index,
    - t2 refers to the normalized phylogenetic mark correlation function or the normalized phylogenetic Simpson index
    - t3 refers to the spatially explicit Simpson index (beta diversity)
    - t4 refers to the ISAR function,
    - t5 refers to the PISAR function,
    - t0 refers the rISAR function, and
    - t6 refers to phylogenetic co-occurrence function



Select summary function

○ phylog. Simpson index
○ norm. phylog. Simpson index
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy



Select summary function

○ phylog. Simpson index
○ norm. phylog. Simpson index
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy

- Zaehler11: the number of point pairs at distance *r* used for t1, t2, t3 and t6 (the denominator of the estimator equation (3.84) in Wiegand and Moloney (2014).
- MCF12_t0, MCF121_t1, …, MCF12_t5, MCF12_t6: the values of different corresponding “bivariate” multivariate summary functions.
- Zaehler12: the number of bivariate point pairs at distance *r* used for t1, t2, t3 and t6.

Additionally, lines 8 and 9 of the *.res files contain the information on the normalization constants for the ISAR and PISAR function needed to estimate the rISAR. For example, line 8 shows for a univariate analysis the following information:

```
number points of foc/count of pattern 1 =   1038 /  11691
exp phl dist11=    0.27212
SD phl dist11=     0.2125
ISAR11_exp=    10.00000
PISAR11_exp=         3.13300
Simpson11_exp=  0.91129170
```

where the number of focal points of pattern 1 (the first number: 1038 in the example) differs from the total number of points of the first multivariate pattern (the second number: 11691 in the example) if you select an individual analysis of a given focal species. In this case it is the number of points of the focal species.

The “`exp phl dist11`” refers to the normalization constant c_{d} of the univariate phylogenetic mark correlation function $k_{\mathrm{d}}(r)$ (i.e., the mean phylogenetic distance between all pairs of individuals) if you exclude conspecifics and it refers to the normalization constant D^{P} of the univariate phylogenetic Simpson index $\beta_{\mathrm{phy}}(r)$ if you include conspecifics. The “`SD phl dist11`” is the corresponding standard deviation.

The “`ISAR11_exp`” and “`PISAR11_exp`” are the normalization constants of the *ISAR* and *PISAR*, respectively, which are their corresponding asymptotes. They are needed to normalize the *rISAR* function (`rISAR11_exp = PISAR11_exp/ISAR11_exp`).

The “`Simpson11_exp`” is the non-spatial Simpson index if you include conspecifics, otherwise it has a value of 1.

7.6.2 Aggregation formulas

The aggregation formulas combine the results of the summary functions of individual replicate analyses into a single aggregated summary function. To this end they use the additional information on the number of pairs of points or the number of points of the focal species (for individual analyses) saved in the *.rep files. The aggregation procedure consists of three steps. In a first step we combine the non-normalized summary functions, in a second step we combine the normalization constants, and finally, the normalized functions [i.e., $k_{\mathrm{d}}(r)$, $\beta_{\mathrm{phy}}(r)$, and $PISAR_f(r)$ and $rISAR_f(r)$] are obtained by dividing the aggregated non-normalized summary functions by the aggregated normalization constants.

There is one smaller difference between null models randomizing the dissimilarity matrix and null models that keep the dissimilarity matrix but randomize the location of the individuals. Because the dissimilarity matrix is randomized each simulation of the null model, we have to estimate aggregated normalization constants for each simulation of the null model separately. However, since randomizing the locations of the individuals does not change the dissimilarity matrix, the normalizations constants do not change during the simulations of the null model.

Aggregation formulas for non-normalized summary functions

We have two groups of estimators that work for several summary functions of the same family in the same way, one for the Simpson family summary functions (that are based on second-order summary functions) and one for the *ISAR* family summary functions (that are based on nearest neighbor summary functions).

For example, the estimator of the spatially-explicit phylogenetic Simpson index is given by

$$\beta_{phy}(r) = \frac{\sum_{i,j,\neq} \delta_{ij}^{P} I(\|x_i - x_j\| - r)}{\sum_{i,j,\neq} I(\|x_i - x_j\| - r)} = \frac{En(r)}{De(r)}$$

where x_i and x_j are the coordinates of the *i*th and *j*th point and the indicator function $I(\|x_i - x_j\| - r)$ =1 if $\|x_i - x_j\| < r$ and $I(\|x_i - x_j\| - r)$ = 0 otherwise. Thus, the denominator $De(r) = \sum_{i,j} I(\|x_i - x_j\| - r)$ counts the number of pairs of points of the multivariate pattern that are distance *r* apart.

The aggregation formula now takes advantage of the ratio nature of the estimator of $\beta_{\text{phy}}(r)$. Because we have for one plot *p* the estimator $\beta_{\text{phy},p}(r) = En_p(r)/De_p(r)$ we generalize the ratio estimator to *P* plots and obtain:

$$\beta_{phy}(r) == \frac{En_1(r) + ... + En_P(r)}{De_1(r) + ... + De_P(r)}$$

See Wiegand and Moloney (2014), section 3.2.1 and equation 3.107 for more detail on the underlying principles of this type of aggregation formulas. The same aggregation formula applies also for the other non-normalized summary functions of the Simpson family, including $c_{\text{d}}(r)$, $C_{\text{d}}(r)$, $\alpha_{\text{phy}}(r)$, $\beta_{\text{S}}(r)$, $\beta_{\text{phy}}(r)$ and $c_{\text{phy}}(r, I)$ and its individual versions (for one focal species). The formula applies also to the individual versions of the summary functions.

The estimator of the non-normalized *PISAR* function for plot *p* is given by

$$PISAR_{f,p}(r) = \sum_{j=1}^{S} \delta_{fj}^{P} D_{fj,p}(r)$$

were the summary function $D_{fj,p}(r)$ gives the proportion of individuals of the focal pattern *f* that have a point of pattern *j* within distance *r*. This suggests an aggregation formula that is the weighted average of the *PISAR* functions of plots *p*, weighted with the relative abundances f_p of the focal species *f* among all plots (section 3.2.1.2 in Wiegand and Moloney 2014):

$$PISAR_f(r) = \sum_{p=1}^{P} f_p PISAR_{f,p}(r) = \sum_{p=1}^{P} f_p \sum_{j=1}^{S} \delta_{fj}^{P} D_{fj}(r) .$$

Note that you can also treat the analyses of the different focal species *f* as “replicates” and in this case the aggregation formula is identical to the community level average *PISAR* function

$$\overline{PISAR}(r) = \sum_{f=1}^{S} f_f PISAR_f(r) = \sum_{f=1}^{S} f_f \sum_{i=1}^{S} D_{fj}(r) \delta_{fj}^{P} .$$

Aggregation formulas for normalized summary functions

Because the normalization constants represent the value of the corresponding summary function without spatial structure, we can obtain aggregation formulas for the normalization constants based on the same principle as used for the summary functions.

For the summary functions of the Simpson family that are based on pairs of points [i.e., $c_\mathrm{d}(r)$, $C_\mathrm{d}(r)$, $\alpha_\mathrm{phy}(r)$, $\beta_\mathrm{phy}(r)$ and $c_\mathrm{phy}(r, I)$] we can rewrite the normalization constants, as exemplified for the phylogenetic Simpson index D^P in the following way:

$$D^P = \sum_{i=1}^{S} f_i \sum_{j=1}^{S} \delta_{ij}^P f_j = \frac{\sum_{i=1}^{S} n_i \sum_{j=1,\neq}^{S} \delta_{ij}^P n_j}{\sum_{i=1}^{S} n_i \sum_{j=1,\neq}^{S} n_j} = \frac{En}{De}$$

and obtain again a ratio where the denominator gives the total number of pairs of points, being

- n (n-1) for the case of one multivariate pattern (i.e., “univariate”) and
- n_f (n - 1) in case of the summary function for an individual focal species f,
- n_1 n_2 for two multivariate patterns.
- n_f n_2 for two multivariate patterns but for an individual focal species f

Thus, the aggregation formula to combine the normalization constant

$D_p^\mathrm{P} = En_p/De_p = En_p$ /[(n_p (n_p - 1)]

of P different plots p is given by:

$$D^P = \frac{[n_1(n_1-1)]D_1^P(r) + \ldots + [n_P(n_P-1)]D_P^P}{[n_1(n_1-1)] + \ldots + [n_P(n_P-1)]} = \sum_{p=1}^{P} w_p D_p^P$$

were the weights w_p are the relative number of pairs of points in each plot.

The aggregation formula for the normalization constant $S_f^\mathrm{P} = \sum_{j=1}^{S} \delta_{fj}^\mathrm{P}$ of the *PISAR* is then given in analog to the aggregation formula for the *PISAR* by

$$S_f^\mathrm{P} = \sum_{p=1}^{P} f_p S_{f,p}^\mathrm{P}$$

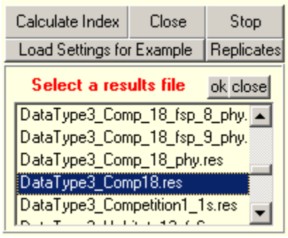
where the weights f_p are the relative abundance of the focal species f among all plots p. The final normalized combined summary function is the combined non-normalized summary function divided by the combined normalization constant.

7.6.3 Example for aggregation formulas

The example file “DataType3_Com_18.phy” is one of the null communities of Shen et al. (2013) with intra- and interspecific competition among individuals separated by less than 5m. Each species pair (*a*, *b*) was assigned an index of ecological similarity *sim*(*a*, *b*) that was randomly drawn from a uniform distribution between 0 and 1. To generate communities with phylogenetic spatial structure driven by competition, a multitype Strauss point process was used where the strength of competition between two species was positively correlated (R^2_{adj} > 0.95) with their ecological similarity. Thus, more similar species tended to locally exclude each other, and as a consequence, phylogenetic evenness was expected to occur for plants located at distances below 5m (i.e. the range of direct competition) (scenario c6 in Table 1 of Shen et al. 2013). The community comprises ten species.

To conduct the analyses for the ten species individually and then combine them into community level summary functions follow this sequence of actions:

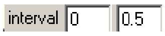
1. Read the settings file “”DataType3_Comp_18.res in window Select a results file and click the small ok button.

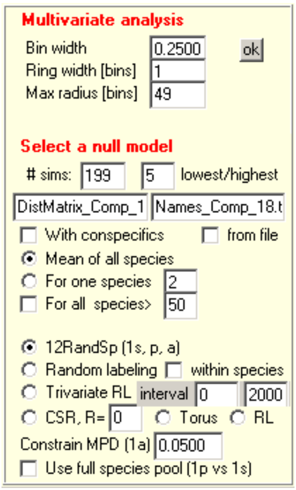


Select a results file
DataType3_Comp_18_fsp_8_phy.
DataType3_Comp_18_fsp_9_phy.
DataType3_Comp_18_phy.res
DataType3_Comp18.res
DataType3_Competition1_1s.res

2. *Programita* now reads all setting for the community level analysis.

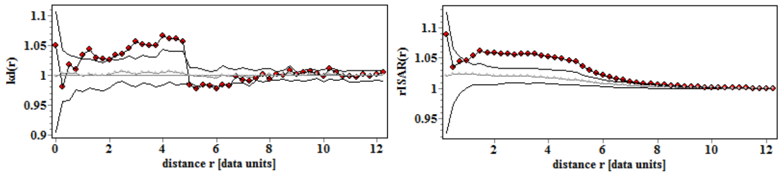
3. For the ring width select a value of *dr* = 1. This allows you for more flexibility in changing the ring width and using the cumulative functions when combining the replicates. You may also select for the phylogenetic co-occurrence function an interval of 0 to 0.5 (dissimilarities range between 0 and 1:


interval 0 0.5

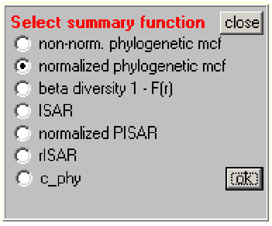


Multivariate analysis
Bin width 0.2500
Ring width [bins] 1
Max radius [bins] 49
Select a null model
# sims: 199 5 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
□ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50
● 12RandSp (1s, p, a)
○ Random labeling □ within species
○ Trivariate RL interval 0 2000
○ CSR, R= 0 ○ Torus ○ RL
Constrain MPD (1a) 0.0500
□ Use full species pool (1p vs 1s)

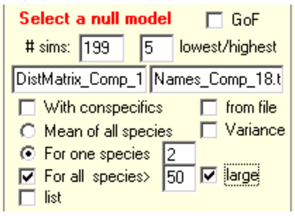
4. Click “**Calculate index**” and *Programita* runs one analysis with the species shuffling null model at the community level. This are the results for the phylogenetic mark correlation function and the *rISAR* function



The phylogenetic mark correlation function shows that individuals with distances below 5m are neighbored by individuals that a more dissimilar than expected by chance (i.e., the species shuffling null model). This result was expected by construction of the community. The cumulative *rISAR* shows a similar result, but here the significant departures have a somewhat larger range.



Select summary function
○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy

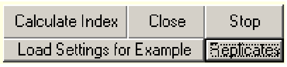


Select a null model □ GoF
# sims: 199 5 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
□ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 2
☑ For all species> 50 ☑ large
□ list

5. Now run the individual analyses for all focal species. To this end select “For one species” and enable the check box “For all species >”. Because all species have more than 50 individuals, leave the value of 50. Additionally, enable the check box “large” to save both, the *.res and *.rep files. The latter contain all information needed to combine the replicates.

6. Click Calculate Index to run the individual analyses.

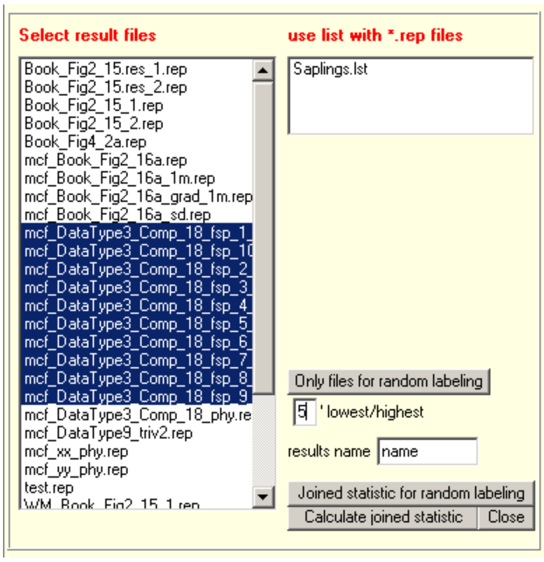
7. Once all simulations are finished, close *Programita* and open it again. Click the button “Replicate” and a window for combining replicates opens:



8. Highlight the files you want combine, they follow the name convention `mcf_DataType3_Comp_18_fsp_nr_phy.rep` with nr being the n umber of the focal species, here running from 1 to 10. The name is based on the name of the data file `DataType3_Com_18.phy.`

9. Give the “rule” for the simulation envelopes, it is the 5th lowest and highest of 199 simulations of the species shuffling null model. Thus, write “5”.

10. Click the button “Calculate joined statistics” and *Programita* combines the summary functions of the ten focal species:



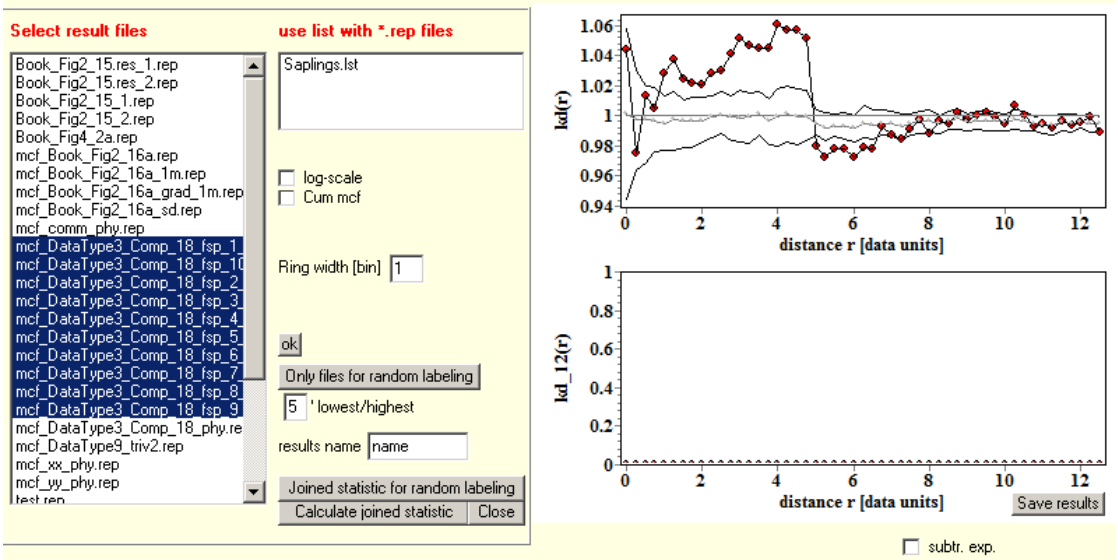
Select result files

Book_Fig2_15.res_1.rep
Book_Fig2_15.res_2.rep
Book_Fig2_15_1.rep
Book_Fig2_15_2.rep
Book_Fig4_2a.rep
mcf_Book_Fig2_16a.rep
mcf_Book_Fig2_16a_1m.rep
mcf_Book_Fig2_16a_grad_1m.rep
mcf_Book_Fig2_16a_sd.rep
mcf_DataType3_Comp_18_fsp_1
mcf_DataType3_Comp_18_fsp_10
mcf_DataType3_Comp_18_fsp_2
mcf_DataType3_Comp_18_fsp_3
mcf_DataType3_Comp_18_fsp_4
mcf_DataType3_Comp_18_fsp_5
mcf_DataType3_Comp_18_fsp_6
mcf_DataType3_Comp_18_fsp_7
mcf_DataType3_Comp_18_fsp_8
mcf_DataType3_Comp_18_fsp_9
mcf_DataType3_Comp_18_phy.re
mcf_DataType9_triv2.rep
mcf_xx_phy.rep
mcf_yy_phy.rep
test.rep
WM_Book_Fig2_15_1.rep

use list with *.rep files

Saplings.lst

Only files for random labeling
5 ' lowest/highest
results name name
Joined statistic for random labeling
Calculate joined statistic Close



Select result files

Book_Fig2_15.res_1.rep
Book_Fig2_15.res_2.rep
Book_Fig2_15_1.rep
Book_Fig2_15_2.rep
Book_Fig4_2a.rep
mcf_Book_Fig2_16a.rep
mcf_Book_Fig2_16a_1m.rep
mcf_Book_Fig2_16a_grad_1m.rep
mcf_Book_Fig2_16a_sd.rep
mcf_comm_phy.rep
mcf_DataType3_Comp_18_fsp_1
mcf_DataType3_Comp_18_fsp_10
mcf_DataType3_Comp_18_fsp_2
mcf_DataType3_Comp_18_fsp_3
mcf_DataType3_Comp_18_fsp_4
mcf_DataType3_Comp_18_fsp_5
mcf_DataType3_Comp_18_fsp_6
mcf_DataType3_Comp_18_fsp_7
mcf_DataType3_Comp_18_fsp_8
mcf_DataType3_Comp_18_fsp_9
mcf_DataType3_Comp_18_phy.re
mcf_DataType9_triv2.rep
mcf_xx_phy.rep
mcf_yy_phy.rep
test.rep

use list with *.rep files

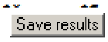
Saplings.lst

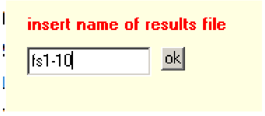
□ log-scale
□ Cum mcf

Ring width [bin] 1

ok
Only files for random labeling
5 ' lowest/highest
results name name
Joined statistic for random labeling
Calculate joined statistic Close

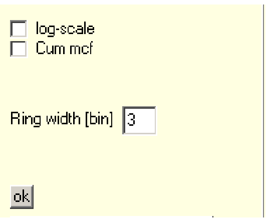
11. To save the results of the combined analysis click the check box “Save results” that appears in the bivariate results graph and



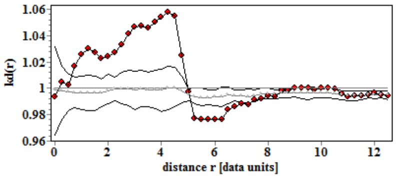

insert name of results file

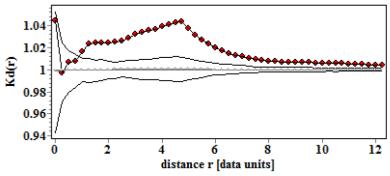
ls1-10

12. You can now change the ring width or obtain the corresponding cumulative summary function. For enlarging the ring width provide for example a value of 3 and click the small “ok” button and for obtaining the cumulative function click “Cum mcf”:

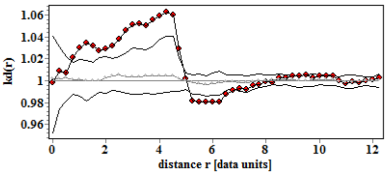

□ log-scale
□ Cum mcf

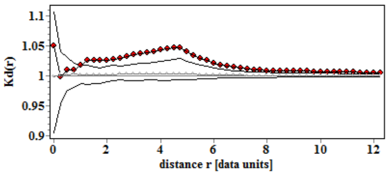
Ring width [bin] 3



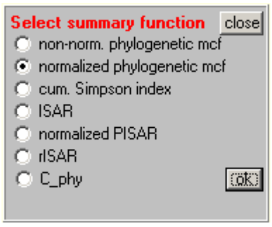


13. The simulation envelopes of the combined summary functions are slightly narrower compared with that of the community level analysis:





This is because the individual analyses across replicate plots do not have the same randomization of the dissimilarity matrix, but the community wide analysis de facto synchronizes the randomization of the dissimilarity matrix for each focal species. Therefore, more extreme cases of the normalization constants that may occur during the community wide analysis are “buffered”.


Select summary function

○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ cum. Simpson index
○ ISAR
○ normalized PISAR
○ rISAR
○ C_phy

8 Multivariate analysis with null communities

8.1 Null communities for one multivariate pattern

In the previous chapter we analyzed multivariate data sets to **detect small-scale spatial correlations in the dissimilarities of neighbored individuals**, independent of the overall functional or phylogenetic community structure, based on variants of the species shuffling null model that randomizes the dissimilarity matrix. In this case the locations of all individuals (as well as the species membership of the individuals) do not change.

The point pattern **null community approach** is fundamentally different because it **randomizes the locations of the individuals of the community** (but not the dissimilarity matrix) by applying point pattern null model techniques that keeping certain features of the patters unchanged, but randomizing others. It allows to simulate spatially-explicit “null communities” that maintain the observed richness, the relative abundances of all and the pairwise dissimilarities between species (i.e., regional species-pool effects are removed), but **randomize tree locations** within each plot following spatial point-process models that resemble different (null) hypotheses on the presence or absence of mechanism such as dispersal limitation, habitat filtering, and interspecific species interactions (Wiegand and Moloney 2014; Wang et al., 2015, 2016, 2018). Thus, this approach can be used to **test the relative importance of local mechanisms of species assembly**. The summary functions of the simulated null community patterns are then compared with that of their observed counterparts (Shen et al. 2009, Wang et al. 2011, Wiegand and Moloney 2014). This allows for comparison of the support obtained by the different alternative null communities from the data using e.g., log-likelihood functions and the Akaike information criterion.

Five types of spatially explicit null communities are mostly used (e.g., Shen et al. 2009; Wang et al. 2015, 2018):

- the random-placement hypothesis
- the habitat-filtering hypothesis
- the dispersal-limitation hypothesis
- the combined habitat and dispersal hypothesis
- the independent placement hypothesis

The construction principle of all these null communities is similar. To generate one realization of a null community, the following steps are conducted. **First**, for each species *s* present in the plot a map is generated were the observed individuals are relocated following stochastic rules that correspond to the hypothesis underlying the null community. For example, the random placement null community represents the extreme case of a community without spatial structure where all species are completely randomly distributed and pairwise independent. In this case, the individuals of species *s* are relocated to random locations in the plot. **Second**, the randomized patterns of all species *s* are joined to obtain one realization of the null community. That means that all null communities conserve the observed plot-level species richness and relative abundances of species and are assembled by independent superposition of the species null model patterns. This corresponds to the assumption of no species interactions (Wiegand et al., 2012). **Finally**, this procedure is repeated as many times as realizations of the null community are desired (e.g., 199 or 999). The algorithms of the specific point process models have been described in detail by Wiegand and Moloney (2014) and Wang et al. (2015, 2018).

These null communities

1) maintain always the observed species abundances,

2) maintain the observed dissimilarity matrix,

3) show always **pairwise independence between species**. Thus, the null communities are assembled without considering the effects of species interactions,

4) can maintain for each species s the **observed larger-scale intensity function** $\lambda_s(x)$ that can be estimated parametrically (e.g., a log-linear regression model that determines the probability that a given small area *xdx* contains an individual of species s) or non-parametrically (e.g., by using a kernel estimate of the intensity function),

5) can maintain for each species *s* the **observed small-scale aggregation**. This can be done parametrically (e.g., by fitting a Thomas process to each species pattern and then using realizations of the Thomas process) or non-parametrically by using pattern-reconstruction (Wiegand et al. 2013). I provide a **software for pattern reconstruction** that generates multiple realizations of patterns that show the same summary functions as the observed pattern [e.g., $g(r)$, $L(r)$, $H_s(r)$, $D_1(r)$, $D_2(r)$,…]. The files are already saved in a format to be read by *Programita* via the "from file" null model option. They can be directly used for the individual analyses e.g., by using the *ISAR* or *rISAR* function, or for the analyses on the community level they must be combined with the files of all other species into a *.phy file. Note that pattern reconstruction can also consider an intensity function $\lambda_s(x)$ and therefore reconstruct patterns in a way that they reproduce faithfully the large-scale (intensity) and small-scale aggregation structures. For more detail on pattern reconstruction see Wiegand et al. (2013) and Wiegand and Moloney (2014).

The elements 3) to 5) can be combined into the five types of spatially explicit null communities mentioned above and described in more detail below. In principle it is also possible to use more complex null community models that incorporate species interactions (e.g., Strauss or Gibbs processes; Illian et al. 2008) or other mechanism. Ultimately even refined point pattern **null community approaches are limited** in detecting processes because they are static and do not describe the underlying dynamical (spatial) mechanisms and processes directly. An alternative is **fitting dynamic and individual-based community simulation models to point pattern summary functions** (e.g., May et al. 2015, 2016), using methods of statistical inference for stochastic simulation models (e.g., Hartig et al. 2011, Lehmann and Huth 2015).

8.1.1 The random-placement hypothesis

This hypothesis assumes that all individuals in the study area are randomly and independently distributed. Thus, it fulfils properties 1) - 3) shown above, but it does not conserve the observed larger-scale intensity function (property 4) and it does not conserve the observed small-scale aggregation structure (5). This null community represents the **extreme case of communities without spatial structure** that do therefore not show distance dependence in spatial community dissimilarity and its components. Note that the summary functions of the Simpson family of the random placement hypothesis are given by the corresponding normalization constants.

To implement the random-placement hypothesis, *Programita* uses for each species *s* a homogeneous Poisson process model with the observed intensity λ_s (CSR) that assigns each tree a random location within the given study area *W*. Significant deviations from this null model indicate existence of non-random spatial structures in spatial community diversity.

8.1.2 The habitat-filtering hypothesis

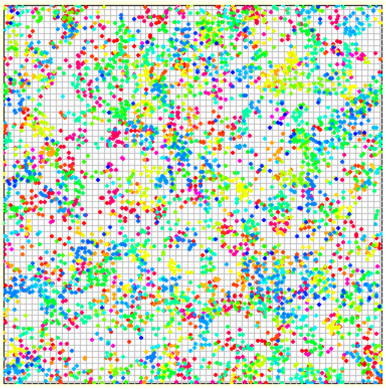
This hypothesis assumes that the **distribution of each species is only driven by local habitat suitability**, but all further mechanism of species patterning are removed. Thus, the habitat filtering null communities fulfil properties 1) - 3) shown above, do **conserve the observed larger-scale intensity function** (property 4), but do not conserve the observed small-scale aggregation structure (5).

To test the habitat filtering hypothesis, a **habitat model** must be derived for each species *s* that relates the observed species locations to environmental variables. The habitat model then provides the intensity function λ_s(x) to be used in an inhomogeneous Poisson process model to generate the realizations of the null model patterns for each species *s*. If a species does not show significant relationships with environmental variables (or if it contains too few observations), the constant plot-scale density λ_s can be used instead of λ_i(x). Parametric intensity functions must be estimated outside *Programita*.

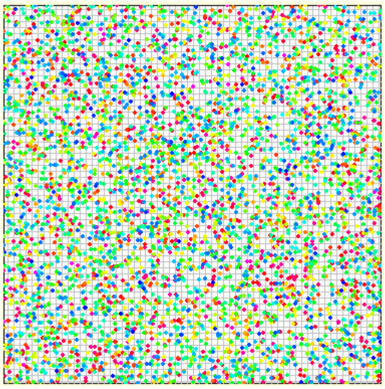
The inhomogeneous Poisson process produces species patterns where the local density of individuals is proportional to the local habitat suitability given by λ_s(x), but no additional mechanisms of species aggregation (i.e., property 5 above) are considered. Significant deviations from this null model therefore indicate that mechanism and processes beyond habitat filtering are operating. However, departures from this hypothesis may also be caused by missing environmental variables or extinction/recolonization dynamics where not all suitable areas are occupied by the species.

A simplified version of the habitat filtering null communities based on a **non-parametric intensity estimate** is implemented in *Programita*. In this case each individual is assigned a random position within a neighborhood of distance *R* of its observed location. This is equivalent to a (box) kernel estimate with bandwidth *R* and yields a heterogeneous Poisson process. If the habitat changes typically at larger scales, a location distance *R* away from the original location will most likely show similar habitat conditions. Thus, local random displacement of the individuals may remove effects caused by small-scale species interactions but place the individuals within similar habitats. This implementation of the habitat filtering null community requires an estimate of the bandwidth *R* which should be larger than the distance where species interactions (e.g., competition or facilitation) operate, but below the typical scale of habitat change (i.e., a separation of scales). Examples and more justification for this approach are given in Wiegand and Moloney (2014).

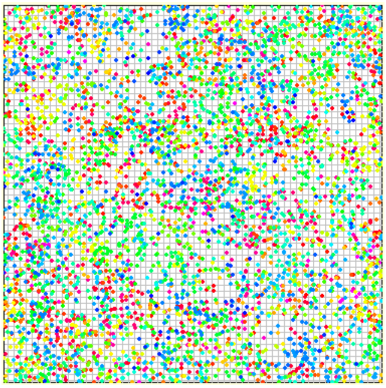
Observed
Dispersal limitation community



Random placement null community
homogeneous Poisson



Habitat filtering null community,
approximated with heterogeneous
Poisson, *R* = 20



8.1.3 The dispersal limitation hypothesis

This hypothesis assumes that the community is **only assembled by the effects of intraspecific aggregation** or other mechanisms of population dynamics (e.g., dispersal limitation or negative conspecific density dependence) without consideration of the influences of habitat filtering or interspecific species interactions. Thus, the dispersal limitation null communities fulfil properties 1) - 3) shown above, do not conserve the observed larger-scale intensity function (property 4), but **conserve the observed univariate small-scale aggregation structure of all species** (5).

To implement this hypothesis, the flexible non-parametric homogeneous pattern reconstruction method should be used (although you could also use parametric methods of fitting cluster point processes). Pattern reconstruction is based on a non-parametric annealing algorithm and able to create for each species null-distribution patterns that closely match the spatial structure of the original pattern as captured by summary functions such as the pair correlation function, the *K*-function and the kth nearest neighbour functions (for detail see Wiegand et al. 2013). Note that the dispersal limitation hypothesis null communities are not implemented in *Programita*, the null model files for each species must be generated separately. The pattern reconstruction output files are already saved in a format to be read by *Programita* via the “from file” null model option. They can be directly used for the individual analyses e.g., by using the *ISAR* or *rISAR* function, but for the analyses on the community level they must be combined with the files of all other species into a *.phy file.

Homogeneous pattern reconstruction does not preserve the spatial intensity function λ_s(x) of species *s*, but it preserves the observed overall aggregation (that can be co-determined by habitat filtering). Significant deviations from this null model indicate that habitat filtering and/or interspecific species interactions contribute to the observed patterns.

A simplified version of the dispersal limitation null communities based toroidal shifts is implemented in *Programita*. Each species pattern is shifted a random vector and locations outside the observation window are relocated based on torus geometry. This null community conserves the univariate aggregation structure of each individual species (given that edge effects due to the toroidal shifts are not too strong), but breaks their mutual dependence.

8.1.4 The combined habitat and dispersal hypothesis

This hypothesis assumes that the community is driven by **the joint effects of habitat filtering and dispersal limitation**. Thus, the combined habitat dispersal limitation null communities fulfil properties 1) - 3) shown above, and conserves both, the observed larger-scale intensity function (property 4) and the observed univariate small-scale aggregation structure of all species (5). Null communities resembling this hypothesis are best created like those generated by the dispersal-limitation hypothesis, but the relocation of individuals of species *s* is additionally constrained by the spatial intensity function λ_s(x) used in the habitat filtering hypothesis. This can be done by using heterogeneous pattern reconstruction (Wiegand et al., 2013).

Significant deviations from this null model may result from unmeasured environmental factors that are ignored in the (log-linear regression) habitat models and by interspecific species interactions that are not considered (because the individual species patterns are independently superimposed).

8.1.5 The independent-placement hypothesis

The independent placement hypothesis tests for **local interspecific interactions** by randomizing species independently of another while preserving the overall intraspecific aggregation and the observed larger-scale distribution [i.e., the observed intensity function λ_i(x)]. Thus, individuals of different species are placed at smaller scales without regard of each other (McGill, 2010). Thus, the independent-placement null communities fulfil properties 1) - 3) shown above, and conserves both, the observed larger-scale intensity function (property 4) and the observed univariate small-scale aggregation structure of all species (5). In this the hypothesis is very similar to the combined habitat and dispersal hypothesis, but it uses instead of the parametric intensity estimate **the observed non-parametric intensity estimate**.

Therefore, to test this hypothesis, one can use the method of the combined habitat and dispersal hypothesis, but a non-parametric kernel estimate of $\lambda_i(x)$ with bandwidth *R* (Wiegand et al., 2013) replaces the parametric estimate. The non-parametric estimate basically smoothes the observed distribution pattern and therefore faithfully reproduces the observed larger scale variation in local tree density. Significant deviations from this null model can therefore only happen at distances *r* smaller than the bandwidth *R*, and mainly as a result of local interspecific species interactions (or imperfect pattern reconstructions or small-scale edaphic factors).

8.1.6 "Univariate" null communities implemented in *Programita*

I implemented several simple null communities in *Programita* that allow you for quick checks of spatial structure in multivariate data sets:

- CSR (random placement)
- a non-parametric heterogeneous Poisson process (habitat filtering)
- a global toroidal shift (dispersal limitation)
- a local toroidal shift (combined habitat and dispersal limitation)
- local and global random labeling

These null communities can be accessed in the "multivariate analysis" null model window via ○ CSR, R= 20 ● Torus ○ RL .

However, more refined null communities such as that presented in Wang et al. (2015, 2018) need to be generated outside of *Programita* and uploaded with the "from file" option.

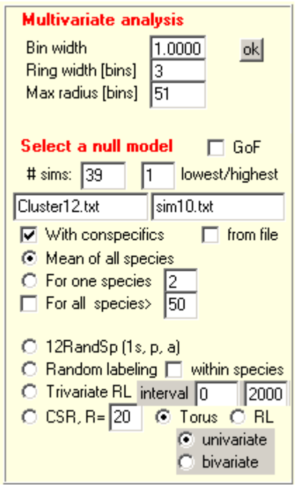
All null communities maintain the observed abundances of the different species in the observation window.

CSR (random placement)

The **CSR** null community ● CSR, R= 0 (with parameter $R = 0$) assumes that all individuals of each species in the study area are randomly and independently distributed, and represents the extreme case of communities without spatial structure. It implements the homogeneous Poisson process for each species.

Non-parametric heterogeneous Poisson process

The **CSR** null community with ● CSR, R= 30 (with parameter $R > 0$) moves each individual to a random position within distance R around the original location. It thereby maintains the larger scale distribution pattern of the species (that will be mainly caused by habitat association), but randomizes small-scale structures (that may be caused by species interactions). This null community therefore implements for each species a simple representation of the inhomogeneous Poisson process with a non-parametric intensity function based on a box kernel with bandwidth R.



Multivariate analysis

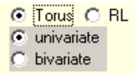
Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 51

Select a null model □ GoF
# sims: 39 1 lowest/highest
Cluster12.txt sim10.txt
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

○ 12RandSp (1s, p, a)
○ Random labeling □ within species
○ Trivariate RL interval 0 2000
○ CSR, R= 20 ● Torus ○ RL
● univariate
○ bivariate

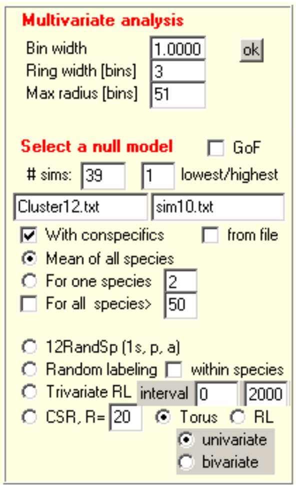
Toroidal shift

The **Torus** null community ● Torus together with ● univariate and R= 0 conducts for each species a toroidal shift (i.e., all individuals of a specie are displaced the same random vector and those landing outside the observation window are moved with torus geometry back to into the observation window). The toroidal shift null communities maintain the clustering of individual species, but remove potential spatial associations among species and potential associations of species to habitat. The **Torus** null community ● Torus together with ○ bivariate applies only for data sets with two multivariate patterns (e.g., that of large trees and that of small trees) and moves the first multivariate pattern as a whole relative to the second one which is maintained.



● Torus ○ RL
● univariate
○ bivariate

Local toroidal shift



Multivariate analysis

Bin width 1.0000 ok

Ring width [bins] 3

Max radius [bins] 51

Select a null model □ GoF

sims: 39 1 lowest/highest

Cluster12.txt sim10.txt

☑ With conspecifics □ from file

● Mean of all species

○ For one species 2

□ For all species> 50

○ 12RandSp (1s, p, a)

○ Random labeling □ within species

○ Trivariate RL interval 0 2000

○ CSR, R= 20 ● Torus ○ RL

● univariate

○ bivariate

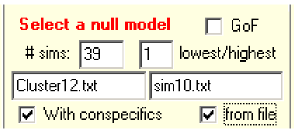
The **local toroidal shift** null community R= 20 ● Torus (with parameter $R > 0$) together with ● univariate conducts for each species a local toroidal shift (i.e., all individuals of a specie are displaced the same random vector (with length < R) and those landing outside the observation window are moved with torus geometry back to into the observation window). The local toroidal shift null communities maintain the clustering of individual species and because each individual is moved not more than distance R of its original location, it also maintains in approximation potential associations to larger-scale habitat. However, this null community remove potential spatial associations among species and potential associations of species to habitat. Because the species patterns are moved entirely with torus geometry, stronger edge effects my appear.

Random labeling null communities

The **RL** null community together with R= 0 conducts random labeling of the species label. It is not really appropriate as null community model, but provided for completeness.

For values of $R > 0$ the **RL** null community keeps the total density of all individuals at scale R (based on a non-parametric intensity estimate with a box kernel and bandwidth R) but places each species following a heterogeneous Poisson process based on this intensity. In this null community the observed habitat association of individual species is not maintained, but it keeps the total density.

From file



Select a null model □ GoF

sims: 39 1 lowest/highest

Cluster12.txt sim10.txt

☑ With conspecifics ☑ from file

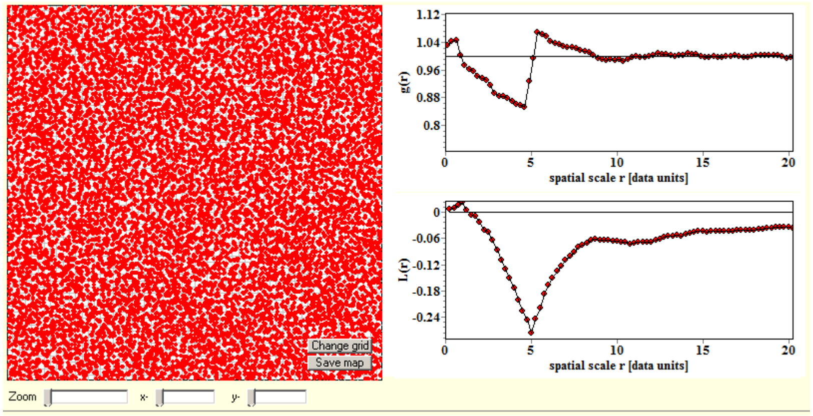
This option allows you to read null community or null model files that were created outside of *Programita*, for example by using the pattern reconstruction software. For individual analyses, e.g., based on the *ISAR* family where only the individuals of the focal species are relocated, you can use the pattern reconstruction output files directly. However, for analyses on the community level you need to join the individual species files of one realization into one *.phy null community file.

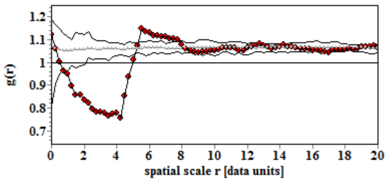
8.2 Analysis of one multivariate pattern based on null communities

The first example file “DataType3_Com18.phy” is one of the null communities of Shen et al. (2013) with intra- and interspecific competition among individuals separated by less than 5m. Each species pair (*a*, *b*) was assigned an index of ecological similarity *sim*(*a*, *b*) that was randomly drawn from a uniform distribution between 0 and 1. To generate communities with phylogenetic spatial structure driven by competition, a multitype Strauss point process was used where the strength of competition between two species was positively correlated (R^2_{adj} > 0.95) with their ecological similarity. Thus, more similar species tended to locally exclude each other, and as a consequence, phylogenetic evenness was expected to occur for plants located at distances below 5m (i.e. the range of direct competition).

The univariate spatial structure of this null community (file DataType3_Com18_uni.dat) is quite interesting and unusual (left below; DataType3_Com18_uni.res). If we conduct a univariate analysis of all individuals in the community, regardless of species, we find that the neighborhood density (measured by the pair correlation function) declines up to distances of *r* = 5m (due to competition), but shows afterwards a discontinuous jump (middle below) just outside the competition range. This behavior results in V-shape of the *L* function (middle).

If we analyze each species separately and then use combine replicates to combine the univariate pair correlation functions to one average univariate function we find the same type of hyperdispersion (right) as for the pattern of all individuals together. The particular shape of the pair correlation and *K* functions is therefore mostly caused by the intraspecific competition of the multitype Strauss point process.





8.2.1 Data preparation for analysis based on null communities

For the multivariate null community analysis you need three (or four) types of data sets:

1. data files of the observed community and the null communities with the location and species identity of all individuals. This is an ASCII file with *.phy extension.
2. a data file with the species acronyms and the species numbers. This is an ASCII file with *.txt extension
3. a data file with the dissimilarity matrix. This is an ASCII file with *.txt extension.
4. if the null communities or null model files were created outside *Programita*, you need additionally the *.phy null community files (for community level analysis) and the *.dat focal species null model files (for individual analysis e.g., with the *ISAR* family)

1) The multivariate data files must be an ASCII file with *. phy extension and have the following format:

```
0 300 0 300 14473
113.13   128.45   1   9
186.36   206.32   1   5
 53.40   152.03   1   2
…
```

where

- the first line gives the dimension of the plot (300 × 300 units) in the example and the total number of points in the list (14,473 in the example)
- the first two columns are the coordinates of the points,
- the third column gives the “pattern” (always “1” for “univariate” as in the example, but can be “1” or “2” for other data types as described above,
- the forth column gives the species identifier (being an integer running from 1 to *S*).

2) The file with species numbers and species acronyms (here file sim10.txt) is a **tab delimited** ASCII file with the *.txt extension and the following format:

```
1  SPECI1
2  SPECI2
3  SPECI3
4  SPECI4
5  SPECI5
6  SPECI6
7  SPECI7
8  SPECI8
9  SPECI9
10 SPEC10
```

The first column is the species number and the second a six letter species acronym.

3) The data file with distance matrix (here **Competition_18.txt**) which is a **tab delimited** ASCII file with the *.txt extension and the following format:

```
SPECI1 SPECI2 0.0894
SPECI1 SPECI3 0.5710
SPECI1 SPECI4 0.2303
SPECI1 SPECI5 0.4659
SPECI1 SPECI6 0.0799
SPECI1 SPECI7 0.4373
SPECI1 SPECI8 0.3786
…
```

where the first two columns are the 6 letter species acronyms of the species pair and the third column is the distance between the two species. **Note that this file must be tab or space delimited** and that **the species acronyms in the distance matrix and the species list must exactly match**.

4a) The null community files

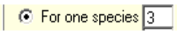
The null community files must have the same format as the *.phy data files. They must also have the same number of individuals and the same species abundance distribution. The name of the data file and the null model files must be the same, except the number at the end:

DataType3_Com18_0.phy (observed data of the community)
DataType3_Com18_1.phy (first null community)
DataType3_Com18_2.phy (second null community)

….
DataType3_Com18_39.phy (last null community).

4b) The null model files for the focal species

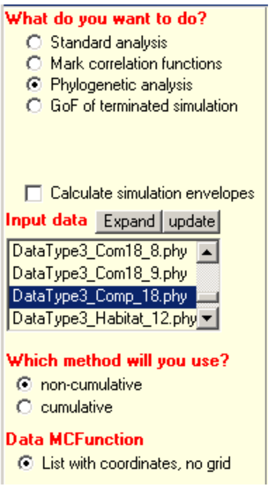
The null model files of the focal species must be standard univariate *.dat files with the same number of individuals as the observed focal species. If you use the pattern reconstruction software, the output files are already in the right format. If the original file of the focal species is named SPECI3.dat, the reconstructed files will be named
rec_SPECI3_1.dat
rec_SPECI3_2.dat
rec_SPECI3_3.dat
….
rec_SPECI3_n.dat

were the number of the simulations of the null model run from 1 to n. The advantage of this name convention is that you only need to change the number of the focal species in the small text box  ● For one species 3 and *Programita* automatically assembles the correct names of the null model files. However, the null model files of the focal species must exist!

8.2.2 Multivariate analysis with "from file" null communities

The **multivariate analysis** based on null community files can be accessed with the following sequence of actions:

1. Select "**Phylogenetic analysis**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data (DataType3_Com18.phy)
3. Click "**List with coordinates, no grid**" in Data MCFunction
4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting "**cumulative**" in Which method will you use. However, this somewhat slows down the estimation and there the better option for estimating the cumulative index is by using the **Replicate** option.
5. Provide in the window Multivariate analysis the bin width in data units (here use 0.25 to have a better spatial resolution since the competition effect occurs at distances below 5m), an appropriate ring width (here 5 bins = 1.25m), and a maximal distance *r* of the analysis (here 80bin = 20m). However, you can also use a ring width of one and increase the ring width later using the **Replicates** option.



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

□ Calculate simulation envelopes

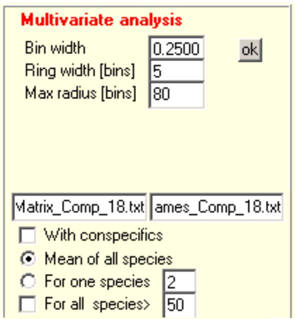
Input data

DataType3_Com18_8.phy
DataType3_Com18_9.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
● List with coordinates, no grid

6. Write the names of the distance matrix (DistMatrix_Comp_18.txt) and the species list (Names_Comp_18.txt) into boxes
7. For the standard univariate analysis select "**Mean of all species**".
8. You can include or exclude the focal species with the check box **With conspecifics.** For the phylogenetic Simpson index use option "With conspecifics", for the phylogenetic mark correlation function disable this option.

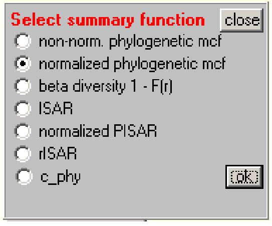


Multivariate analysis

Bin width 0.2500
Ring width [bins] 5
Max radius [bins] 80

Matrix_Comp_18.txt
ames_Comp_18.txt
□ With conspecifics
● Mean of all species
○ For one species 2
□ For all species> 50

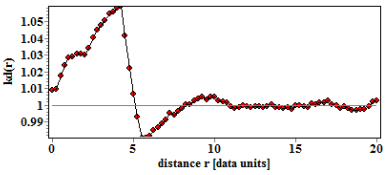
9. Click "**Calculate index**" and *Programita* estimates the different summary functions of the data. Select them in the window Select one test function:



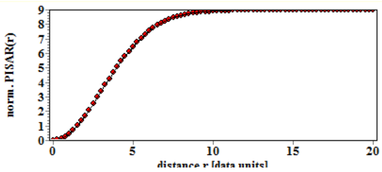
Select summary function

○ non-norm. phylogenetic mcf
● normalized phylogenetic mcf
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy

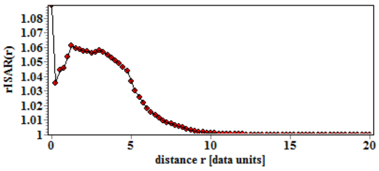
phylog. mark correlation function



normalized *PISAR*



rISAR



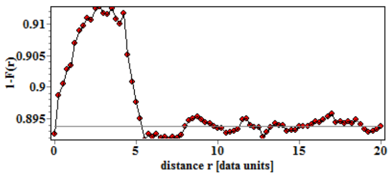
The phylogenetic mark correlation function shows a clear signal of competition, up to distance *r* = 5m: there is phylogenetic evenness (i.e., species are neighbored by more dissimilar neighbors). However at distances larger than 5 m we observe a tendency to phylogenetic clustering.

The jump in the phylogenetic mark correlation function can be explained by frequent cases where three plants are arranged linearly as ABC where the distance between A and C is just outside the range of competition (say 6–10 m). Because species pairs AB and BC will be dissimilar (due to competition), the pair AC may show a small phylogenetic distance. This explains the tendency towards phylogenetic clustering just outside the range of competition which disappears smoothly with increasing spatial distance. The *PISAR* function shows that phylogenetic diversity saturates only at about 10m whereas the *rISAR* functions suggests that individual species are surround within neighborhoods of 10m by phylogenetically more dissimilar species.

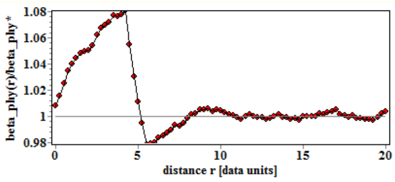
10. You can also select the option “**With conspecifics**” access the spatially explicit Simpson index (i.e., beta diversity) and the phylogenetic Simpson index:

☑ With conspecifics

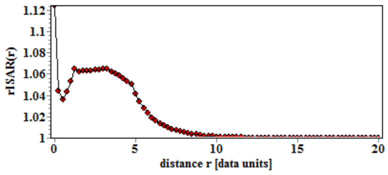
Simpson index $\beta(r)$



phylogenetic Simpson index $\beta_{\mathrm{phy}}(r)$



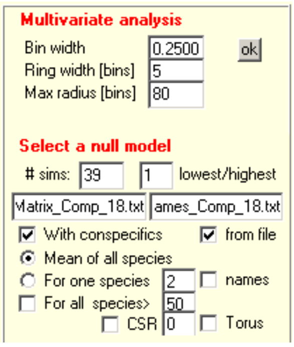
rISAR



The Simpson index shows that proportion of heterospecifics has a peak of approximately 92% at about 3-5m and then dramatically declines to values of about 89.5% at distances larger than 5m. The phylogenetic Simpson index is very similar to the phylogenetic mark correlation function, and the *rISAR* is little affected by inclusion or not of conspecifics.

11. Click “**calculate simulation envelopes**”

12. Select the number of simulations of the null model (39) and the rule for the simulation envelopes (1), and select for the univariate analysis of null communities the null model “**from file**”. Usually 199 null communities (or more) are recommended.



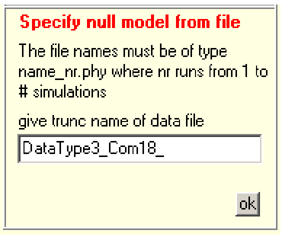
Multivariate analysis

Bin width 0.2500 ok
Ring width [bins] 5
Max radius [bins] 80

Select a null model

sims: 39 1 lowest/highest
Matrix_Comp_18.txt ames_Comp_18.txt
☑ With conspecifics ☑ from file
● Mean of all species
○ For one species 2 □ names
□ For all species> 50
□ CSR 0 □ Torus

The “**from file**” null model reads the null model patterns from files which need to have exactly the same structure as the file with the observed data. The name of the data file and the null model files must be the same, except the number at the end:



Specify null model from file

The file names must be of type name_nr.phy where nr runs from 1 to # simulations

give trunc name of data file

DataType3_Com18_

ok

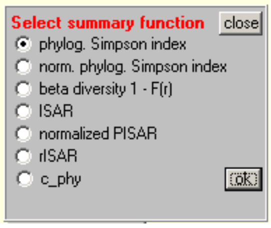
DataType3_Com18_0.phy (data of community)
DataType3_Com18_1.phy (first null community)
DataType3_Com18_2.phy (second null community)

….
DataType3_Com18_39.phy (last null community)

To read the null community files add the trunk name: “DataType3_Com18_” and click the small “ok” button.

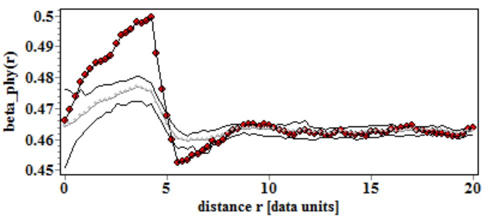
13. The null community in the example corresponds to the "**dispersal limitation hypothesis**" in Shen et al. (2009) and Wang et al. (2015, 2018). It assumes that the community would be assembled only by action of dispersal limitation and other internal mechanisms of population dynamics that can create intraspecific species patterns such as clustering, but that habitat filtering or species interactions do not influence the placement of trees. The species patterns of the null community were created by using non-parametric techniques of pattern reconstruction (Wiegand et al. 2013) to exactly conserve the observed species patterns, but the patterns of the individual species were independently superimposed to remove any signal of species interactions. Thus, this null community does not show effects of smaller-scale species interactions which are incorporated into the data file by the multitype Strauss process.

14. After enable the option "With conspecifics", clicking "**Calculate index**" and running the simulations of the null model you can select with the window Select one test function among different test functions.



Select summary function
● phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy

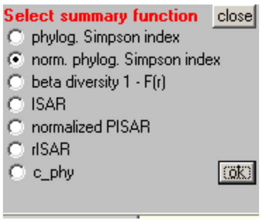
- Select "phylog Simpson index" to see results for the **phylogenetic Simpson index** $\beta_{phy}(r)$



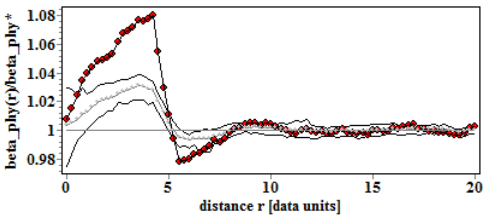
which clearly indicates at distances below 5m higher than expected phylogenetic evenness, phylogenetic clustering at distances just outside the range of competition, and no effect afterwards. Note that the expectation of the $\beta_{phy}(r)$ (grey lines) yields the expected spatially explicit Simpson index $\beta_S(r)$ of the null communities.

Thus, the community shows strong spatial structure caused by the particular univariate structure of the different species, but the observed data contain additional structure in species dissimilarities that does not exist in the null communities.

- Select “norm. phylog Simpson index” to obtain the **normalized phylogenetic Simpson index** $\beta_{phy}(r)/D^{P}$:

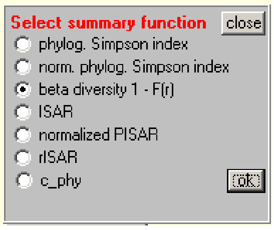


Select summary function close
○ phylog. Simpson index
● norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy
ok

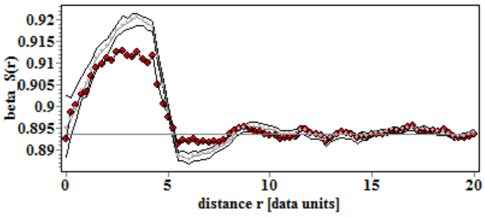


For null communities, the results of the normalized phylogenetic Simpson index differs from that of the non-normalized index only by the normalization constant D^{P}. For larger distances it approximates, as expected, a value of one.

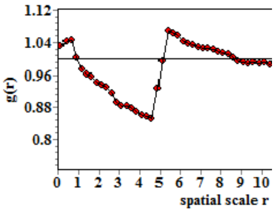
- Select “beta diversity 1 – F(r)” to obtain the spatially explicit Simpson index $\beta(r)$ which yields the probability that two individuals distance r apart are of the same species. This summary function describes the (inverse) **distance decay relationship**:



Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
● beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy
ok

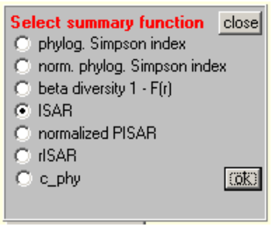


The analysis shows that the null communities that conserve the univariate spatial pattern of each species approximates the observed distance decay relationship, but does not fully explain it. This is because of the negative interspecific species interactions of the Strauss process where species are not independently placed. However, the figure also shows that the largest contribution to species beta diversity stems from the intraspecific competition. The elevated null community values of $\beta(r) = 1 - F(r)$ at distances smaller than 5m are caused by the hyperdispersion of the individual species patterns that allow more distantly related heterospecifics to fill in.



The observed pattern in the distance decay is a consequence of both, intraspecific competition and the hyperdispersion pattern of univariate neighborhood densities up to distances of 5m (i.e., fewer conspecific neighbors than expected).

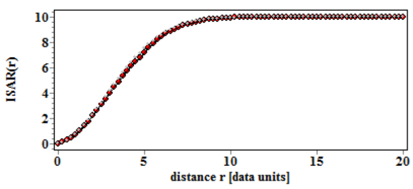
- Select “ISAR” you obtain **the average individual-species area relationship**. Because departures from the null community are not visible in the standard *ISAR* plot use the ☑ subtr. exp. option that subtracts the expectation of the null community:

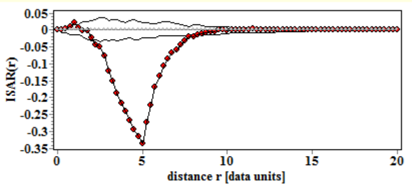


Select summary function close

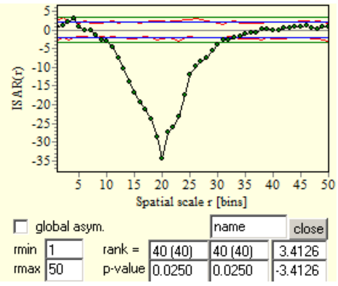
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy

ok





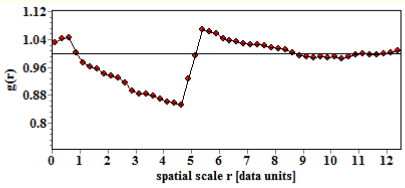
You can also use the GoF test with the student transformation:



☐ global asym. name close

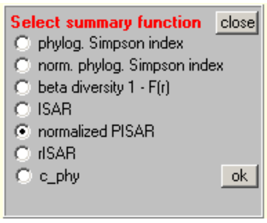
rmin 1 rank = 40 (40) 40 (40) 3.4126

rmax 50 p-value 0.0250 0.0250 -3.4126



The results show that the typical individual of the community is surrounded up to neighborhoods of 5m (= 20 bins) by fewer species than expected by pairwise independence of species (i.e., without small-scale species interactions). This was expected because of the heterospecific competition and the reduced neighborhood density (right graph). For distances larger than 5m this effect fades away (note that the *ISAR* is a cumulative index). Compare this behavior to that of the *L*-function of all points!

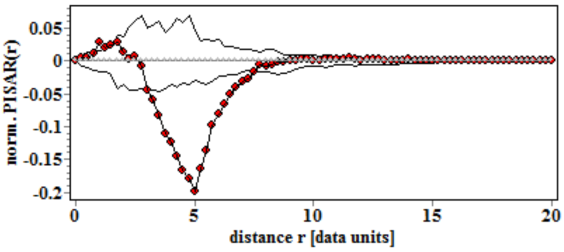
- Select “normalized PISAR” to obtain **the normalized PISAR** function. the ☑ subtr. exp. option that subtracts the expectation of the null community:



Select summary function close

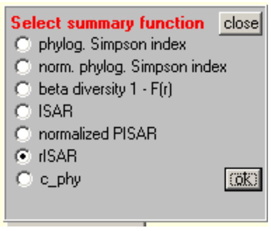
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
● normalized PISAR
○ rISAR
○ c_phy

ok



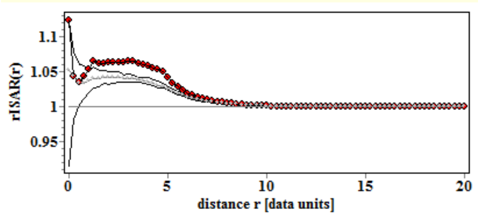
The result is similar to that of the average *ISAR*, but departures from the null community are somewhat weaker. The average *PISAR* is therefore driven to a large extent by the underlying species patterns which are captured by the average *ISAR*.

- Select rISAR to obtain the ***rISAR*** function.

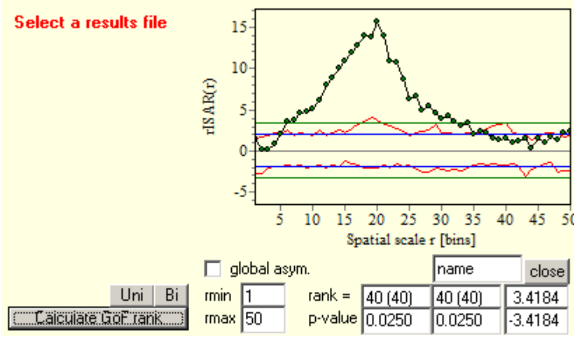


Select summary function

○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
● rISAR
○ c_phy



The *rISAR* indicates that the typical individual of the community is surrounded up to neighborhoods of some 5m by more dissimilar species than expected by pairwise species independence. The GoF test shows that the phylogenetic evenness increases up to distances of 5m (= 20 bins of 0.25m) (due to the competition of more similar species) and then declines smoothly (because it is a cumulative index):



Select a results file

□ global asym.

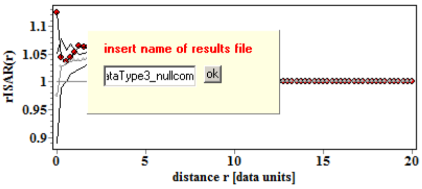
name

rmin 1
rmax 50

rank = 40 (40) 40 (40) 3.4184
p-value 0.0250 0.0250 -3.4184

Thus, phylogenetic diversity increases quicker than species diversity. This was expected because a species was more likely to be surrounded by more dissimilar species.

15. After the analysis is finished save results with the “Save results option”



insert name of results file

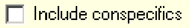
ataType3_nullcom

To save results provide name. The results are saved as file name_phy.res and mcf_name_phy.rep.

16. The *.res file contains all your settings and can be used to load all settings to repeat the analysis (using option “**Load Settings for Example**”.

The *.rep file contains the detailed results of your analysis that allows you to reassemble them. **However, you can only use the full features of this option if the ring width of the analysis was one**.

Phylogenetic mark correlation functions and null communities

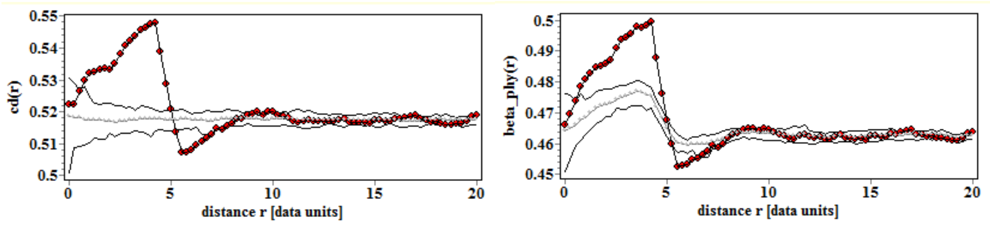
17. You can access the phylogenetic mark correlation function instead of the phylogenetic Simpson index. The difference is that you include or not conspecifics. To access the phylogenetic mark correlation function **disable** the option "With conspecifics". 
□ Include conspecifics

18. After clicking "**Calculate index**" and run the simulations of the null model you can select with the window Select one test function among different test functions:

    a. Select "non-norm. phylogenetic mcf" to obtain the **non- normalized phylogenetic mark correlation function** $c_d(r)$ (i.e., the normalizing constant $c_d = 1$).

non-normalized phylogenetic mark correlation function

non-normalized phylogenetic Simpson index

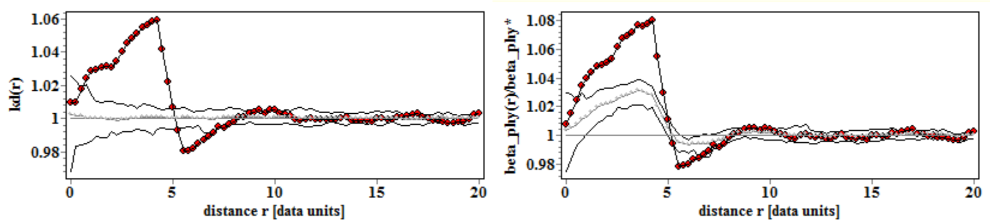


Comparison with the phylogenetic Simpson index shows that the effect of pure species placement (captured by the Simpson index) is factored out and that the expectation of the null community is a constant without distance decay.

    b. Select "normalized phylogenetic mcf" to obtain the **normalized phylogenetic mark correlation function** $k_d(r)$:

normalized phylogenetic mark correlation function

normalized phylogenetic Simpson index



The results of the normalized phylogenetic mark correlation function $k_d(r)$ show that the expectation of the null community yields indeed a value of one.

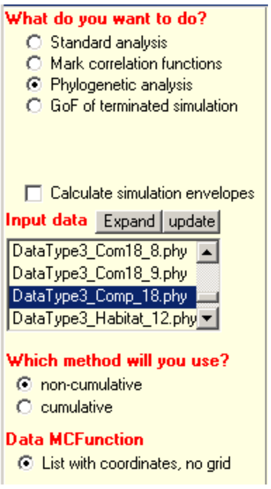
    c. If you select "beta diversity 1 – F(r)" the beta diversity graph appears as for the case where you include conspecifics.

    d. The results of the three summary functions "ISAR", "PISAR" and "rISAR" that include conspecifics are very similar to the results without conspecifics.

8.2.3 Toroidal shift null communities

In the previous example I used the best possible implementation of the dispersal limitation null communities based on pattern reconstruction. Now I show that the shortcut of approximating the dispersal limitation null communities through toroidal shifts produces very similar results. The multivariate analysis based on the toroidal null communities can be accessed with the following sequence of actions:

1. Select "**Phylogenetic analysis**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data (DataType3_Com18.phy)
3. Click "**List with coordinates, no grid**" in Data MCFunction
4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting "**cumulative**" in Which method will you use. However, this somewhat slows down the estimation and there the better option for estimating the cumulative index is by using the **Replicate** option.



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

□ Calculate simulation envelopes

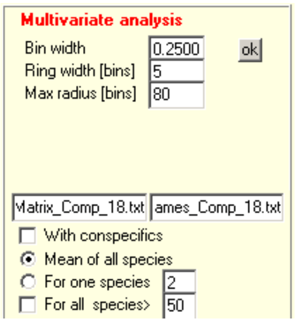
Input data

DataType3_Com18_8.phy
DataType3_Com18_9.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
● List with coordinates, no grid

5. Provide in the window Multivariate analysis the bin width in data units (here use 0.25 to have a better spatial resolution since the competition effect occurs at distances below 5m), an appropriate ring width (here 5 bins = 1.25m), and a maximal distance *r* of the analysis (here 80bin = 20m). However, you can also use a ring width of one and increase the ring width later using the **Replicates** option.
6. Write the names of the distance matrix (DistMatrix_Comp_18.txt) and the species list (Names_Comp_18.txt) into boxes
7. For the standard univariate analysis select "**Mean of all species**".
8. You can include or exclude the focal species with the check box **With conspecifics.** For the phylogenetic Simpson index use option "With conspecifics", for the phylogenetic mark correlation function disable this option.



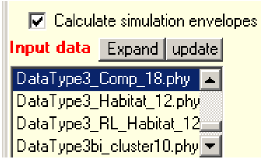
Multivariate analysis
Bin width 0.2500

Ring width [bins] 5
Max radius [bins] 80

Matrix_Comp_18.txt ames_Comp_18.txt
□ With conspecifics
● Mean of all species
○ For one species 2
□ For all species> 50

9. Click "**Calculate index**" and *Programita* estimates the different indices of the data. Select them in the window Select one test function:

10. Select “calculate simulation envelopes” and click in the “Multivariate analysis” window the null model “Torus”.

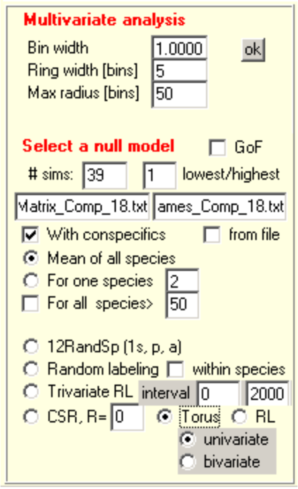


☑ Calculate simulation envelopes

Input data Expand update

DataType3_Comp_18.phy
DataType3_Habitat_12.phy
DataType3_RL_Habitat_12
DataType3bi_cluster10.phy

11. After enable the option “With conspecifics”, clicking “**Calculate index**” and running the simulations of the null model you can select with the window Select one test function among different test functions.



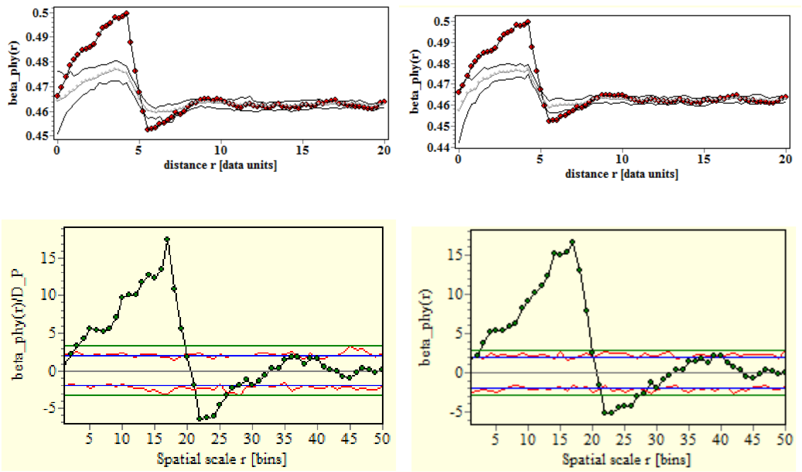
Multivariate analysis

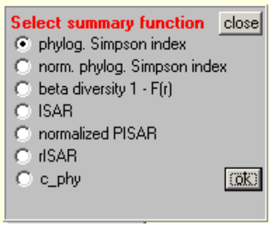
Bin width 1.0000 ok
Ring width [bins] 5
Max radius [bins] 50

Select a null model □ GoF
# sims: 39 1 lowest/highest
Matrix_Comp_18.txt ames_Comp_18.txt
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50
○ 12RandSp (1s, p, a)
○ Random labeling □ within species
○ Trivariate RL interval 0 2000
○ CSR, R= 0 ● Torus ○ RL
● univariate
○ bivariate

12. Select “phylog Simpson index” to see results for the **phylogenetic Simpson index** $\beta_{phy}(r)$

dispersal limitation toroidal shift

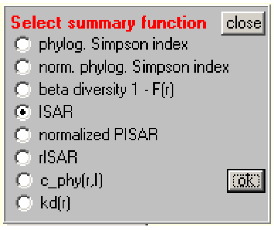




Select summary function close
● phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy
ok

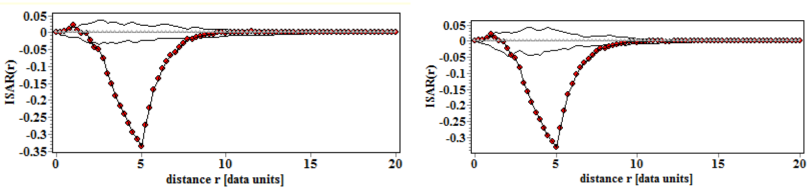
Comparison with the results of the dispersal limitation null communities above show that the toroidal shift communities provide a very good approximation.

13. Select “ISAR” to obtain **the average individual-species area relationship**. Because departures from the null community are not visible in the standard *ISAR* plot use the ☑ subtr. exp. option that subtracts the expectation of the null community:

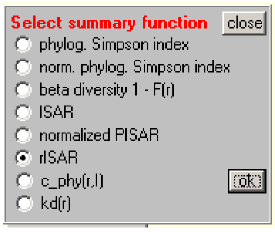


Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy(r,l)
○ kd(r)
ok

dispersal limitation toroidal shift

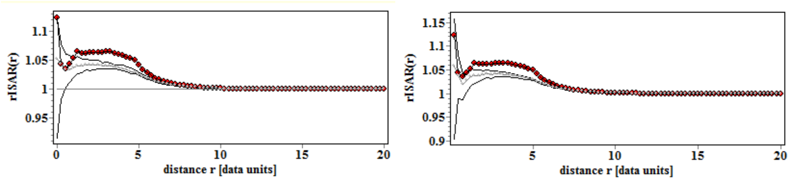


14. Again, there is little difference between results. The same is true for the *rISAR*:

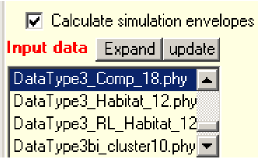


Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
● rISAR
○ c_phy(r,l)
○ kd(r)
ok

dispersal limitation toroidal shift



8.2.4 Local toroidal shift null communities



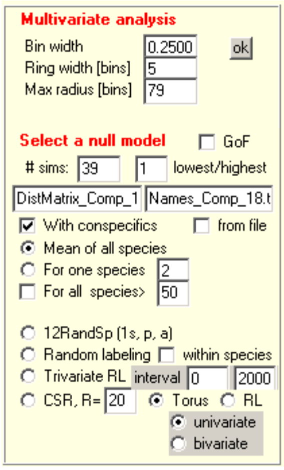
☑ Calculate simulation envelopes

Input data

DataType3_Comp_18.phy
DataType3_Habitat_12.phy
DataType3_RL_Habitat_12
DataType3bi_cluster10.phy

To show an example of the local toroidal shift null communities that approximately maintain the small-scale species aggregation and the larger-scale intensity function, we continue the above example.

1. Repeat the steps 1. - 9. from the DataType3_torus.res example.
2. Select "calculate simulation envelopes" and click in the "Multivariate analysis" window the null model "Torus" and "R = 20m". That means that each species pattern is shifted by a random vector with length < 20 units (i.e., 5m).
3. After enable the option "With conspecifics", clicking "**Calculate index**" and running the simulations of the null model you can select with the window Select one test function among different test functions.



Multivariate analysis

Bin width 0.2500

Ring width [bins] 5

Max radius [bins] 79

Select a null model ☐ GoF

sims: 39 1 lowest/highest

DistMatrix_Comp_1 Names_Comp_18.t

☑ With conspecifics ☐ from file

● Mean of all species

○ For one species 2

☐ For all species> 50

○ 12RandSp (1s, p, a)

○ Random labeling ☐ within species

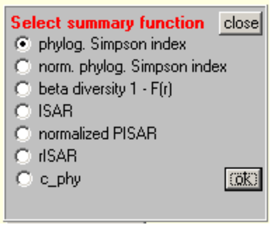
○ Trivariate RL interval 0 2000

○ CSR, R= 20 ● Torus ○ RL

● univariate

○ bivariate

4. Select "phylog Simpson index" to see results for the **phylogenetic Simpson index** $\beta_{phy}(r)$



Select summary function

● phylog. Simpson index

○ norm. phylog. Simpson index

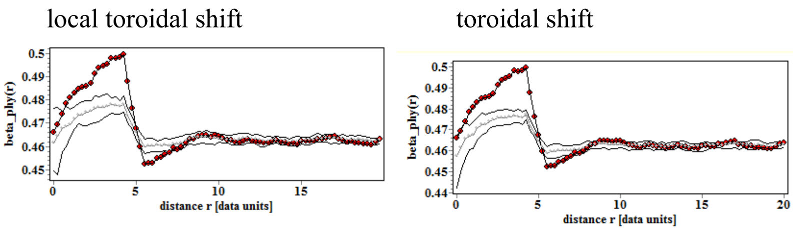
○ beta diversity 1 - F(r)

○ ISAR

○ normalized PISAR

○ rISAR

○ c_phy

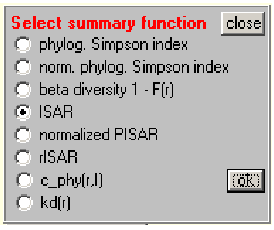


local toroidal shift

toroidal shift

Comparison with the results of the toroidal shift null communities above show very little differences. This shows that the local toroidal shift effectively removes small-scale associations.

5. Select "ISAR" to obtain **the average individual-species area relationship**. Because departures from the null community are not visible in the standard *ISAR* plot use the ☑ subtr. exp. option that subtracts the expectation of the null community:



Select summary function

○ phylog. Simpson index

○ norm. phylog. Simpson index

○ beta diversity 1 - F(r)

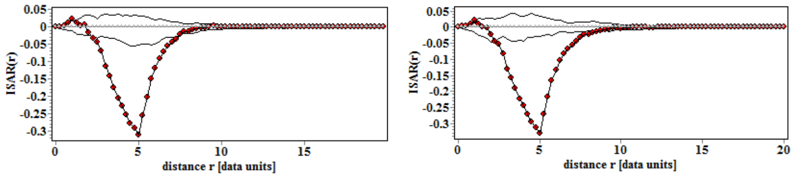
● ISAR

○ normalized PISAR

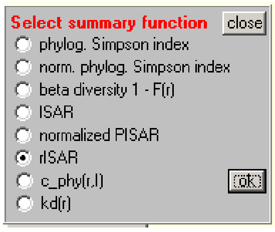
○ rISAR

○ c_phy(r,l)

○ kd(r)



6. Again, there is little difference between results. The same is true for the *rISAR*:



Select summary function

○ phylog. Simpson index

○ norm. phylog. Simpson index

○ beta diversity 1 - F(r)

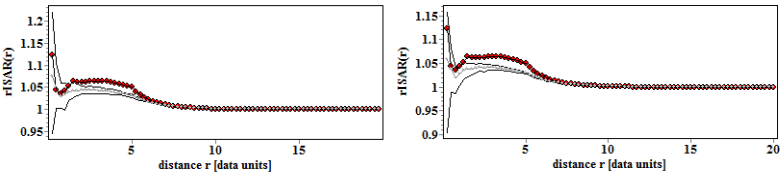
○ ISAR

○ normalized PISAR

● rISAR

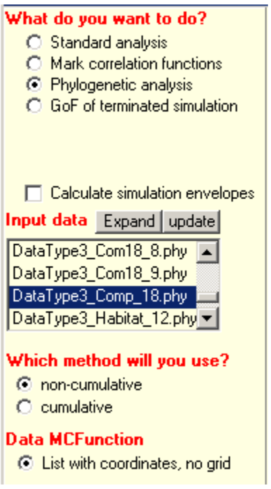
○ c_phy(r,l)

○ kd(r)



8.2.5 Random placement null communities

We now use the data of the previous example to present the random placement null communities. The multivariate analysis based on the random placement null communities can be accessed with the following sequence of actions:



What do you want to do?

○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation

□ Calculate simulation envelopes

Input data

DataType3_Com18_8.phy
DataType3_Com18_9.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy

Which method will you use?

● non-cumulative
○ cumulative

Data MCFunction

● List with coordinates, no grid

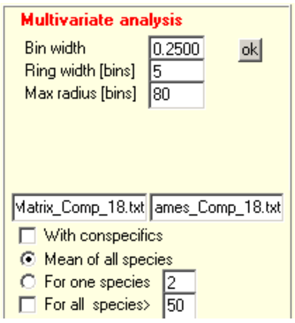
1. Select "**Phylogenetic analysis**" in window What do you want to do?

2. Highlight data file you want to analyze in Input data (DataType3_Com18.phy)

3. Click "**List with coordinates, no grid**" in Data MCFunction

4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting "**cumulative**" in Which method will you use. However, this somewhat slows down the estimation and there the better option for estimating the cumulative index is by using the **Replicate** option.

5. Provide in the window Multivariate analysis the bin width in data units (here use 0.25 to have a better spatial resolution since the competition effect occurs at distances below 5m), an appropriate ring width (here 5 bins = 1.25m), and a maximal distance *r* of the analysis (here 80bin = 20m). However, you can also use a ring width of one and increase the ring width later using the **Replicates** option.



Multivariate analysis

Bin width 0.2500
Ring width [bins] 5
Max radius [bins] 80

Matrix_Comp_18.txt ames_Comp_18.txt

□ With conspecifics
● Mean of all species
○ For one species 2
□ For all species> 50

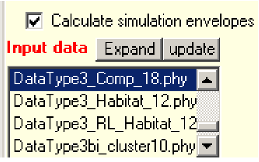
6. Write the names of the distance matrix (DistMatrix_Comp_18.txt) and the species list (Names_Comp_18.txt) into boxes

7. For the standard univariate analysis select "**Mean of all species**".

8. You can include or exclude the focal species with the check box **With conspecifics.** For the phylogenetic Simpson index use option "With conspecifics", for the phylogenetic mark correlation function disable this option.

9. Click "**Calculate index**" and *Programita* estimates the different indices of the data. Select them in the window Select one test function:

10. Select “calculate simulation envelopes” and click in the “Multivariate analysis” window the null model “CSR, R =” with R = 0.



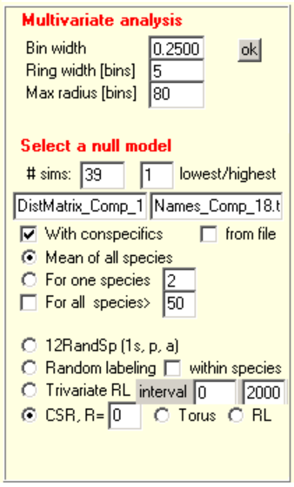
☑ Calculate simulation envelopes

Input data Expand update

DataType3_Comp_18.phy
DataType3_Habitat_12.phy
DataType3_RL_Habitat_12
DataType3bi_cluster10.phy

11. After enable the option “With conspecifics”, clicking “**Calculate index**” and running the simulations of the null model you can select with the window Select one test function among different test functions.

12. Select “phylog Simpson index” to see results for the **phylogenetic Simpson index** $\beta_{phy}(r)$



Multivariate analysis

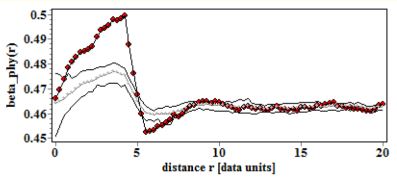
Bin width 0.2500 ok
Ring width [bins] 5
Max radius [bins] 80

Select a null model

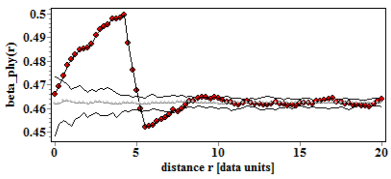
sims: 39 1 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

○ 12RandSp (1s, p, a)
○ Random labeling □ within species
○ Trivariate RL interval 0 2000
● CSR, R= 0 ○ Torus ○ RL

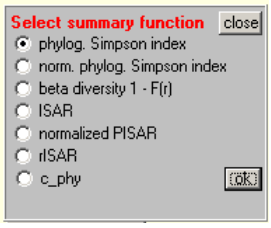
dispersal limitation



random placement

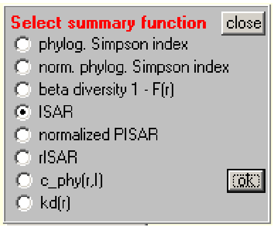


Because the random placement null community does not contain spatial structure, the expectation of the phylogenetic Simpson index $\beta_{phy}(r)$, which is up to a constant the same as the spatially explicit Simpson index $\beta_S(r)$, is a constant and yields the D^P. Comparison with the results of the dispersal limitation null communities show the degree of spatial structure maintained by the toroidal shift communities.

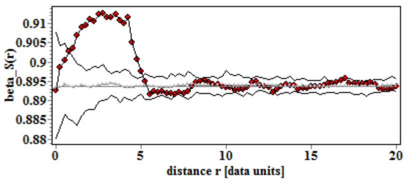


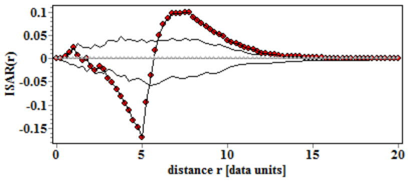
Select summary function close
● phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy ok

13. Select “beta diversity 1 - F(r)” to spatially explicit Simpson index $\beta_S(r)$ and “ISAR” to obtain the average individual-species area relationship. The analysis with the community average ISAR function shows a pattern similar to the neighborhood density with a smaller than expected number of species within the 5m competition zone of influence and a higher number of species just outside the zone of influence.

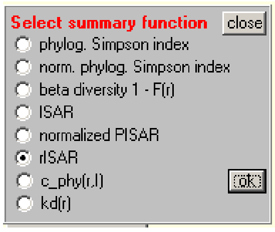


Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy(r,l) ok
○ kd(r)



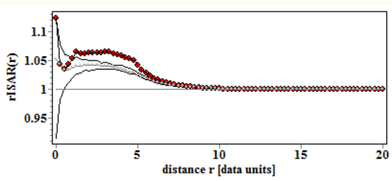


Again, there is large difference in the *rISAR* compared with the dispersal limitation null communities

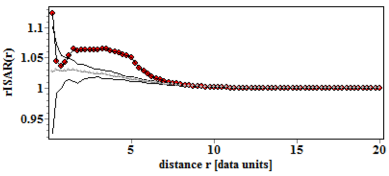


Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
● rISAR
○ c_phy(r,l) ok
○ kd(r)

dispersal limitation



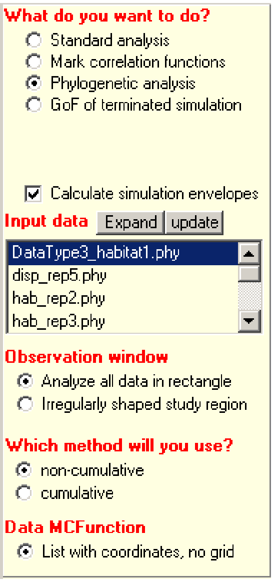
random placement



8.2.6 Local random placement null communities

To show an example of the local random placement null communities I use the example file “DataType3_habitat1.phy”, a simulated data set where 3183 individuals of 100 species are distributed in a 316 × 316 m plot. The community is only structured by larger-scale habitat filtering, but does not show additional effects of species interactions. To generate the data, Wiegand et al. (2017) used the R package metricTester presented in Miller et al. (2017). The multivariate analysis based on the local random placement null communities can be accessed with the following sequence of actions:

1. Select “**Phylogenetic analysis**” in window What do you want to do?
2. Highlight data file you want to analyze in Input data (DataType3_habitat1.phy)
3. Click “**List with coordinates, no grid**” in MCFunction
4. Optionally you can estimate the cumulative spatially-explicit Simpson index (or phylogenetic mark correlation function) by selecting “**cumulative**” in Which method will you use. However, this somewhat slows down the estimation and there the better option for estimating the cumulative index is by using the **Replicate** option.
5. Provide in the window Multivariate analysis the bin width in data units (1m), an appropriate ring width (use ring width of 1 if the analysis takes long time and then the **Replicate** option to change the ring width), and a maximal distance *r* of the analysis (50).



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation
☑ Calculate simulation envelopes

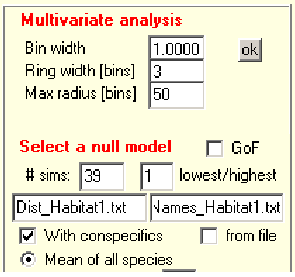
Input data
DataType3_habitat1.phy
disp_rep5.phy
hab_rep2.phy
hab_rep3.phy

Observation window
● Analyze all data in rectangle
○ Irregularly shaped study region

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
● List with coordinates, no grid

6. Provide the file with species numbers and names (here file “Names_Habitat1.txt”)
7. Provide the data file with distance matrix (here Dist_Habitat1.txt)
   For the “univariate” analysis select “**Mean of all species**”

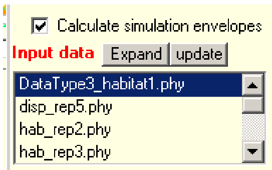


Multivariate analysis
Bin width 1.0000
Ring width [bins] 3
Max radius [bins] 50

Select a null model □ GoF
# sims: 39 1 lowest/highest
Dist_Habitat1.txt Names_Habitat1.txt
☑ With conspecifics □ from file
● Mean of all species

8. You can include or exclude the focal species with the check box **With conspecifics.** For the phylogenetic Simpson index use option “With conspecifics”, for the phylogenetic mark correlation function disable this option.
9. Click “**Calculate index**” and *Programita* estimates the different indices of the data. Select them in the window Select one test function:

10. Select “calculate simulation envelopes” and click in the “Multivariate analysis” window the null model “CSR, R =” with R = 20.

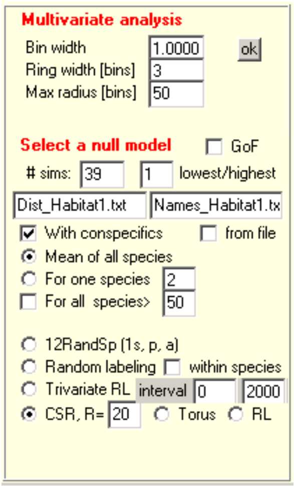


☑ Calculate simulation envelopes

Input data Expand update

DataType3_habitat1.phy
disp_rep5.phy
hab_rep2.phy
hab_rep3.phy

11. After enable the option “With conspecifics” and clicking “**Calculate index**”, *Programita* runs the simulations of the null model. Programita shows on the left the observed community pattern and on the right the null-community patterns. The local random placement null communities maintain the larger-scale species distributions, but randomize smaller-scale structures within 20m:

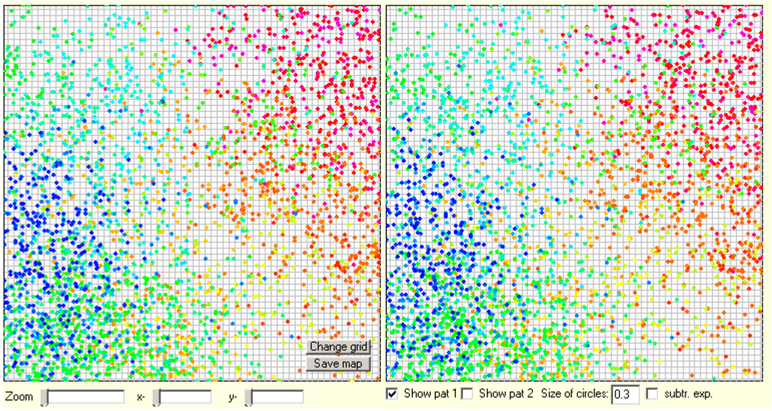


Multivariate analysis

Bin width 1.0000 ok
Ring width [bins] 3
Max radius [bins] 50

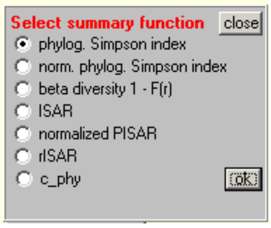
Select a null model □ GoF
# sims: 39 1 lowest/highest
Dist_Habitat1.txt Names_Habitat1.tx
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50

○ 12RandSp (1s, p, a)
○ Random labeling □ within species
○ Trivariate RL interval 0 2000
● CSR, R= 20 ○ Torus ○ RL

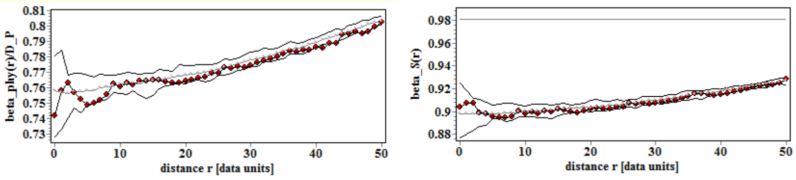


☑ Show pat 1 □ Show pat 2 Size of circles: 0.3 □ subtr. exp.

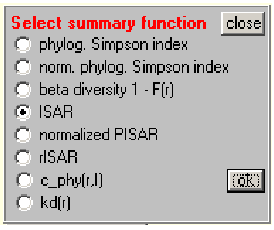
12. Select “phylog Simpson index” and “beta diversity 1 - F(r)”to see results for the phylogenetic Simpson index $\beta_{phy}(r)$ and of the spatially-explicit Simpson index $\beta_S(r)$



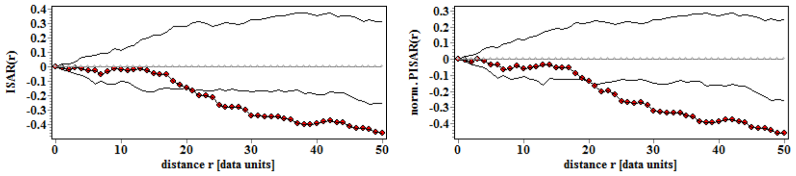
Select summary function close
● phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy ok



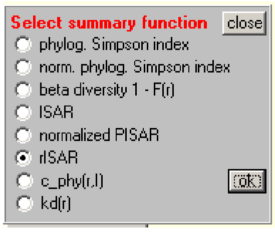
13. However, at larger distances there is a small effect of the *ISAR* and the *PISAR*:



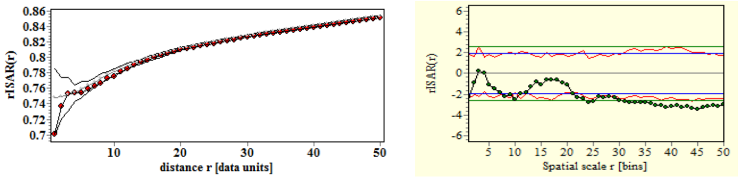
Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy(r,l) ok
○ kd(r)



that is however mostly driven by the *ISAR*, as shown by the *rISAR*:

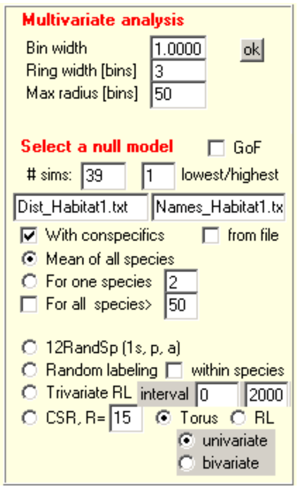


Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
● rISAR
○ c_phy(r,l) ok
○ kd(r)



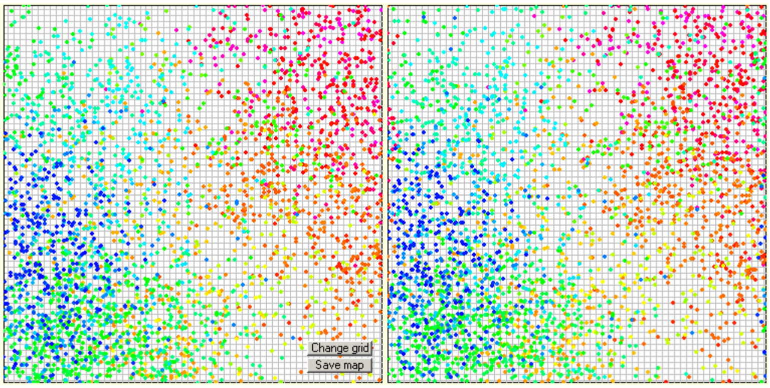
8.2.7 Local toroidal shift null communities 2

To show an example of the local toroidal shift null communities I use the example from above. Repeat steps 1) to 9) and then follow:

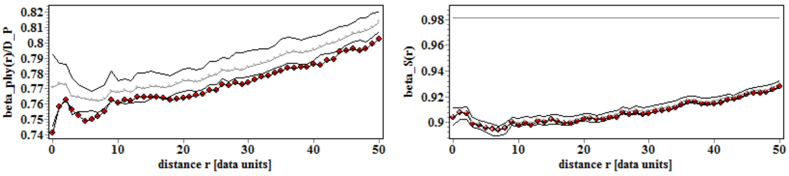


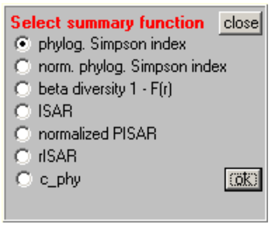
Multivariate analysis
Bin width 1.0000
Ring width [bins] 3
Max radius [bins] 50
Select a null model □ GoF
# sims: 39 1 lowest/highest
Dist_Habitat1.txt Names_Habitat1.tx
☑ With conspecifics □ from file
● Mean of all species
○ For one species 2
□ For all species> 50
○ 12RandSp (1s, p, a)
○ Random labeling □ within species
○ Trivariate RL interval 0 2000
○ CSR, R= 15 ● Torus ○ RL
● univariate
○ bivariate

1. Select “calculate simulation envelopes” and click in the “Multivariate analysis” window the null model “Torus” and R = 15. Now the pattern of each species is moved a random vector with length < 15 units.
2. After enable the option “With conspecifics” and clicking “**Calculate index**”, *Programita* runs the simulations of the null model. Programita shows on the left the observed community pattern and on the right the null-community patterns. The local toroidal shift null communities maintain the larger-scale species distributions, but randomize smaller-scale structures within 15m:



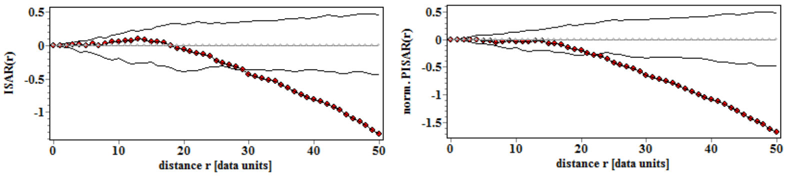
3. The phylogenetic Simpson index $\beta_{phy}(r)$ shows a small departure, but not the spatially-explicit Simpson index $\beta_S(r)$:

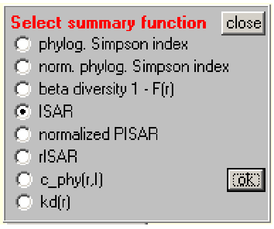




Select summary function
● phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy

4. At larger distances there are smaller effects in the *ISAR* and the *PISAR*:



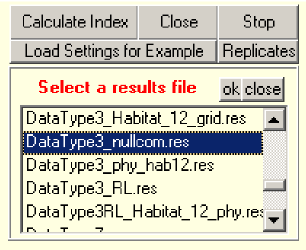


Select summary function
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy(r,l)
○ kd(r)

The departures in the $\beta_{phy}(r)$, *ISAR* and *PISAR* are probably due to the edge effects in the toroidal shift that become visible because the community pattern shows large species patches that are cut in the middle by the border of the observation window.

8.2.8 View results of multivariate analysis with null communities

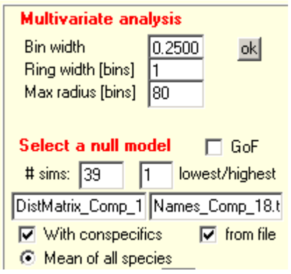
1. Click “**Load Settings for Example**”, select file “*DataType3_nullcom.res*” with the example for the multivariate analysis with null communities, and click the small “ok” button.



Select a results file

DataType3_Habitat_12_grid.res
DataType3_nullcom.res
DataType3_phy_hab12.res
DataType3_RL.res
DataType3RL_Habitat_12_phy.res

2. Select a ring width of 1 in the window Multivariate analysis.



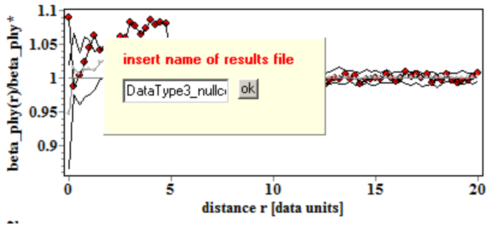
Multivariate analysis

Bin width 0.2500
Ring width [bins] 1
Max radius [bins] 80

Select a null model □ GoF
# sims: 39 1 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
☑ With conspecifics ☑ from file
● Mean of all species

3. Click “**Calculate Index**” to run analysis.

4. Click “Save Results” button and save results under the name “DataType3_nullcom_dr1”:



insert name of results file

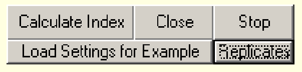
DataType3_nullc

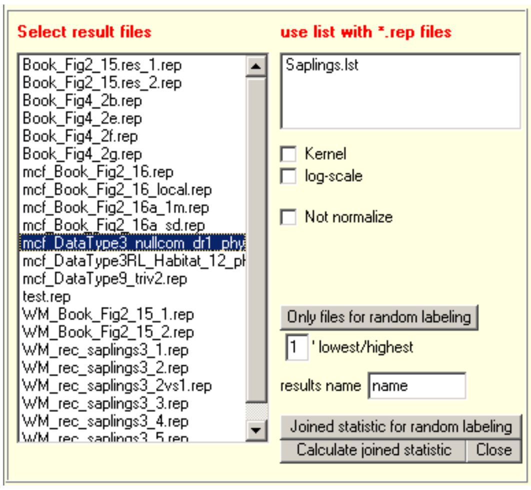
Programita saves the two files
DataType3_nullcom_dr1_phy.res
mcf_DataType3_nullcom_dr1_phy.rep

5. Close and re-open *Programita*.

6. To visualize results or to change the ring width or to make the Simpson indices cumulative, use the “**Replicates**” option.

7. Click “**Replicates**” and select in the window that opens the results file you want to see (mcf_DataType3_nullcom_dr1_phy.rep):




Only files for random labeling
Joined statistic for random labeling
Calculate joined statistic
Close

Select result files

Book_Fig2_15.res_1.rep
Book_Fig2_15.res_2.rep
Book_Fig4_2b.rep
Book_Fig4_2e.rep
Book_Fig4_2f.rep
Book_Fig4_2g.rep
mcf_Book_Fig2_16.rep
mcf_Book_Fig2_16_local.rep
mcf_Book_Fig2_16a_1m.rep
mcf_Book_Fig2_16a_sd.rep
mcf_DataType3_nullcom_dr1_phy
mcf_DataType3RL_Habitat_12_ph
mcf_DataType9_triv2.rep
test.rep
WM_Book_Fig2_15_1.rep
WM_Book_Fig2_15_2.rep
WM_rec_saplings3_1.rep
WM_rec_saplings3_2.rep
WM_rec_saplings3_2vs1.rep
WM_rec_saplings3_3.rep
WM_rec_saplings3_4.rep
WM_rec_saplings3_5.rep

use list with *.rep files

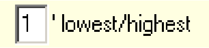
Saplings.lst

□ Kernel
□ log-scale
□ Not normalize

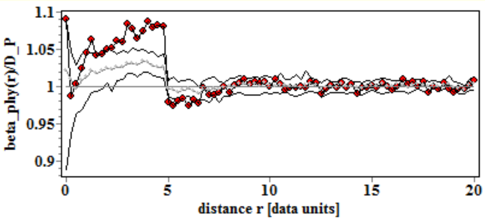
1 ' lowest/highest

results name name

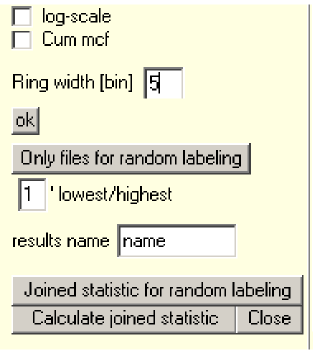
8. Select file mcf_DataType3_nullcom_dr1_phy.rep, select the rule for the simulation envelopes (here the 1th lowest and highest) and then "**Calculate joined statistic**". The results graph is then shown:



1 ' lowest/highest



The graph of the normalized phylogenetic Simpson index looks quite rugged. Select a ring width of 5 and click the small "ok" button:

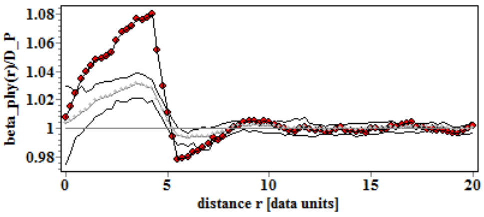


□ log-scale
□ Cum mcf
Ring width [bin] 5

ok
Only files for random labeling

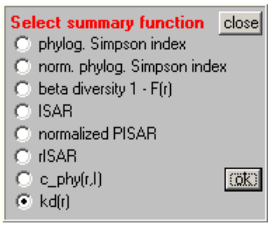
1 ' lowest/highest
results name name

Joined statistic for random labeling
Calculate joined statistic
Close



The result is now much clearer.

9. To select a different summary use the window Select one test function

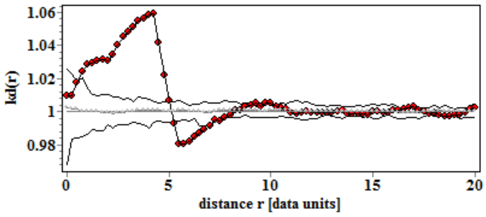


○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy(r,l)
● kd(r)

Here you can additionally select "kd(r)" to obtain the **phylogenetic mark correlation function** that yields

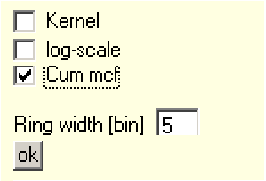
$$k_\mathrm{d}(r) = (1/c_\mathrm{d})\beta_{phy}(r)/\beta(r)$$

where the $c_\mathrm{d} = D^\mathrm{P}/D$ is the normalization constant:



Now the effect of pure species placement (captured by the spatially-explicit Simpson index is factored out) and the $k_\mathrm{d}(r)$ shows how the phylogenetic beta diversity changes relative to the species beta diversity.

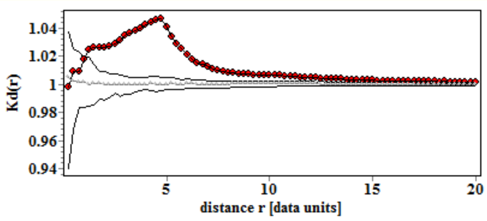
10. To obtain the cumulative version of the summary function clique “Cum mcf” and then the small “ok” button. Now the cumulative phylogenetic mark correlation function is shown:



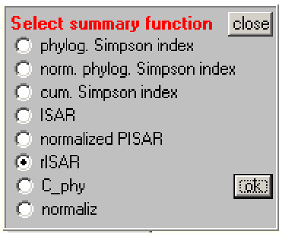
□ Kernel
□ log-scale
☑ Cum mcf

Ring width [bin] 5

ok

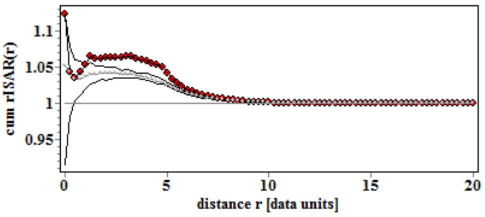


The result of the cumulative function is similar to that of the rISAR function:



Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ cum. Simpson index
○ ISAR
○ normalized PISAR
● rISAR
○ C_phy
○ normaliz

ok



11. Note that you can only change the ring width for the con-cumulative Simpson indices.

12. You can also save the results of the cumulative summary function or the summary function with a different ring width by using the “**Save results**” option.

8.3 Individual analysis of one multivariate pattern

The **individual summary functions** such as the individual species area relationship *ISAR* (Wiegand et al. 2007) or the *rISAR* function (Wang et al. 2016) allow you to conduct the analysis from the viewpoint of the individuals of a given "focal" species. Individual summary functions quantify the biotic neighborhood of the individuals of the focal species by estimating the mean of the species richness, phylogenetic or functional diversity,... etc. of the neighborhoods with radius *r* around the focal individuals. The idea is then to **compare the biotic neighborhood of the focal species to that of random locations in the plot**. Thus, in this case the community of all heterospecifics itself is unchanged, but the locations of the focal species are randomized by a suitable null model. **This is a fundamental difference between the individual analysis and the community level analysis.**

In case of the individual species area relationship *ISAR* (that estimates the expected species richness in the neighborhood with radius *r* around the typical individuals of the focal species; Wiegand et al. 2007) you can explore if the focal species is located in areas of lower or higher than expected species richness. The concept of the individual species area relationship can also be extended to other summary functions of multivariate patterns. For example, using phylogenetic dissimilarity you can explore if the focal species is locally surrounded by ecologically more similar or dissimilar species than expected by the null model.

Programita uses in the individual analysis only the individuals of a given focal species *f* (which are taken from the multivariate focal pattern 1) and counts individuals or species of the multivariate focal pattern 1 or a second multivariate pattern 2. *Programita* estimates therefore always "uni" and "bivariate" summary functions. A **"univariate" analysis** would be for example assessment of the species richness of large trees in the neighborhood of large trees of a focal species, and a **"bivariate" analysis** would be the species richness of small trees in the neighborhood of large trees of a given focal species. However, if the data file does not contain points of a second pattern, only the "univariate" analysis is conducted.

The **individual species area relationship** $ISAR_f(r)$ can be interpreted as the expected species richness within distance *r* of the typical individual of the focal species *f* and is estimated as:

$$ISAR_f(r) = \sum_{j=1}^{S} D_{fj}(r)\delta_{fj}$$

where $D_{fj}(r)$ is the probability that the nearest species *j* neighbor of an individual of the focal species *f* is located within distance *r*, and δ_{fj} is zero for conspecifics (i.e., $f = j$) and one for heterospecifics (i.e., $f \neq j$). However, you can also include the focal species in the count, in this case

$$ISAR_f(r) = \sum_{j=1}^{S} D_{fj}(r)$$

The **individual (spatially-explicit) Simpson index** $\beta_{S,f}(r)$ yields the probability that a point located at distance *r* of a point of the focal species *f* is a heterospecific and is estimated as:

$$\beta_{S,f}(r) = \sum_{j=1}^{S} \delta_{fj} p_{fj}(r) = \sum_{j=1}^{S,\neq} p_{fj}(r)$$

where the $p_{fj}(r)$ are mark connections functions that yield the probability that of two individuals distance *r* apart the first is of type *f* and the second of type *j*.

The **cumulative individual (spatially-explicit) Simpson index** $\alpha_{S,f}(r)$ yields the probability that a randomly selected individual of the community that is located within distance *r* of an individual of the focal species is heterospecific. Note that the index $1 - \alpha_{S,f}(r)$ describes the **local dominance of the focal species** within a neighborhood of radius *r*.

The individual species area relationship and the Simpson index can also be generalized to include instead of the discrete δ_{fj} a continuous measure $\delta^P{}_{fj}$ of (phylogenetic or functional) dissimilarity between the focal species *f* and other species *j*. We obtain the **phylogenetic individual species area relationship *PISAR$_f$*(*r*)** (Wiegand and Moloney 2014):

$$PISAR_f(r) = \sum_{j=1}^{S} D_{fj}(r)\delta_{fj}^P$$

that quantifies the expected phylogenetic (or functional) diversity of species within the neighborhood with radius *r* around the typical individual of the focal species *f*. If the placement of the focal species *f* is unrelated with functional or phylogenetic relationships with their neighbors, the *PISAR* is proportionally to the *ISAR* and fully driven by the local species richness.

To yield a function that is independent on local species richness within the neighborhood *r* we divide the *PISAR* function by the *ISAR* function:

$$rISAR_f(r) = \frac{\sum_{m=1}^{S,\neq} d(f,m)D_{fm}(r)}{\sum_{m=1,}^{S,\neq} D_{fm}(r)}$$

The *rISAR$_f$*(*r*) function therefore yields the expected phylogenetic (or functional) distance between the typical tree of the focal species *f* and all other species within distance *r*. The normalization constant of the *rISAR* function yields $\Delta_f^{\mathrm{P}} = \frac{1}{S-1}\sum_{j=1}^{S} \delta_{fj}^{\mathrm{P}}$, the mean functional (or phylogenetic) dissimilarity between an individual of the focal species *f* and all other species in the plot.

Analogously, the **individual phylogenetic Simpson index** yields

$$\beta_{f,phy}(r) = \sum_{j=1}^{S} \delta_{fj}^P p_{fj}(r)$$

and is the mean pairwise phylogenetic distance taken over all pairs of individuals where the first is of the focal species *f* and the second individual (that can also be of the focal species) is located distance *r* of the first individual. If the placement of the focal species *f* is unrelated with functional or phylogenetic relationships with their neighbors, the individual phylogenetic Simpson index $\beta_{f,\mathrm{phy}}(r)$ is proportionally to the individual spatially-explicit Simpson index $\beta_{f,\mathrm{S}}(r)$ and therefore fully driven by the distance decay in species similarity as quantified by $\beta_{f,\mathrm{S}}(r)$.

To yield a function that is independent on the distance decay in species similarity we divide the $\beta_{\text{phy},f}(r)$ by the $\beta_{\text{S},f}(r)$ to obtain the **individual phylogenetic mark correlation function** $k_{\text{d,f}}(r)$:

$$k_{f,d}(r) = \frac{\beta_{f,phy}(r)}{\beta_{f,S}(r)} = \frac{1}{c_{f,d}} \frac{\sum_{j=1}^{S} \delta_{fj}^{P} p_{fj}(r)}{\sum_{j=1}^{S} \delta_{fj} p_{fj}(r)} = \frac{1}{c_{f,d}} \sum_{j=1}^{S,\neq} \delta_{fj}^{P} p_{fj}(r)$$

The normalization constant $c_{f,\text{d}}$ yields the expected phylogenetic distance of an individual of the focal species and a heterospecific individual taken randomly from the plot.

The normalized **individual phylogenetic mark correlation function** $k_{f,\text{d}}(r)$ yields the expected phylogenetic distance of the typical individual of the focal species *f* to an arbitrary selected heterospecific individual located at distance *r*, relative to its non-spatial expectation $c_{f,\text{d}}$.

8.3.1 Data preparation for analysis based on null communities

For the individual multivariate analysis you need three or four types of data sets:

1. data files of the **observed community**. This is an ASCII file with *.phy extension.
2. a data file with the **species acronyms** and the species identifier (species number). This is an ASCII file with *.txt extension
3. a data file with the **dissimilarity matrix**. This is an ASCII file with *.txt extension.
4. if you use the "from file" option, data files with the **null model locations of the focal species**. This is an ASCII file with the *.dat extension and in the standard format for univariate point pattern analysis.

1) The multivariate data files must be an ASCII file with *. phy extension and have the following format:

```
0 300 0 300 14473
113.13    128.45   1    9
186.36    206.32   1    5
 53.40    152.03   1    2
 75.96    103.34   1    1
 …
```

where

- the first line gives the dimension of the plot (300 × 300 units) in the example and the total number of points in the list (14,473 in the example)
- the first two columns are the coordinates of the points,
- the third column gives the "pattern" ("1" is the point belongs to the focal pattern 1 and "2" if the point belongs to the second patter 2. *Programita* conducts always an **"univariate" analysis** (taking the individuals of the focal species and estimating the neighborhood community properties of pattern 1; e.g., local species richness of large trees around large trees of a given focal species) and if patter 2 exists a **"bivariate" analysis** (taking the individuals of the focal species and estimating the neighborhood community properties of pattern 2; e.g., local species richness of small trees around large trees of a given focal species)
- the forth column gives the species identifier (being an integer running from 1 to *S*).

2) The file with species numbers and species acronyms (here file sim10.txt) is a **tab or space delimited** ASCII file with the *.txt extension and the following format:

```
 1  SPECI1
 2  SPECI2
 3  SPECI3
 4  SPECI4
 5  SPECI5
 6  SPECI6
 7  SPECI7
 8  SPECI8
 9  SPECI9
 10 SPEC10
```

where the first column is the species number and the second column a 6 letter species acronym.

3) The data file with distance matrix (here **Competition_18.txt**) which is a **tab or space delimited** ASCII file with the *.txt extension and the following format:

```
SPECI1 SPECI2 0.0894
SPECI1 SPECI3 0.5710
SPECI1 SPECI4 0.2303
SPECI1 SPECI5 0.4659
SPECI1 SPECI6 0.0799
SPECI1 SPECI7 0.4373
SPECI1 SPECI8 0.3786
SPECI1 SPECI9 0.6580
SPECI1 SPEC10 0.4468
…
```

where the first two columns are the 6 letter species acronyms of the species pair and the third column is the distance between the two species. Note that this file must be tab or space delimited and that the species **acronyms in the distance matrix and the species list must exactly match**.

4) The univariate data files that describe the null model locations of the focal species must be a space or tab delimited ASCII file with the *.dat extension, following the standard format for univariate point pattern analysis:

```
0  300  0  300     1064
 153.95604    69.13256    1   0
  16.77999    28.36763    1   0
 200.69017   247.38376    1   0
  18.00621    63.92836    1   0
 216.78976   147.36916    1   0
  60.95564   141.50468    1   0
...
```

where

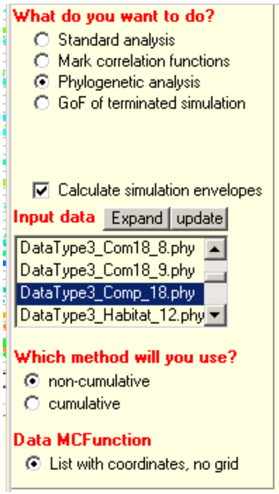
- the first line gives the dimension of the plot (300 × 300 units) in the example and the total number of points in the list (1064 in the example)
- the first two columns are the coordinates of the points,
- the third and forth column must be always “1   0”) for univariate patterns.

The null model files of the focal species must match that of the focal species (of the focal pattern 1) in the *.phy community data file.

These files must have the name “name_n.dat” where “name” is a common string in the names of the files and “n” the number of the null model file that runs from 1 to the # of simulations (e.g., 39 or 199).

8.3.2 *ISAR* and Simpson indices with “from file” option

The example file “DataType3_Com18.phy” already used above for the null community analysis is one of the null communities of Shen et al. (2013) with intra- and interspecific competition among individuals separated by less than 5m. Each species pair (*a*, *b*) was assigned an index of ecological similarity *sim*(*a*, *b*) that was randomly drawn from a uniform distribution between 0 and 1. To generate communities with phylogenetic spatial structure driven by competition, a multitype Strauss point process was used where the strength of competition between two species was positively correlated ($R^2_{\text{adj}} > 0.95$) with their ecological similarity. Thus, more similar species tended to locally exclude each other, and as a consequence, phylogenetic evenness is expected to occur for plants located at distances below 5m (i.e. the range of direct competition) (scenario c6 in Table 1 of Shen et al. 2013).



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation
☑ Calculate simulation envelopes

Input data

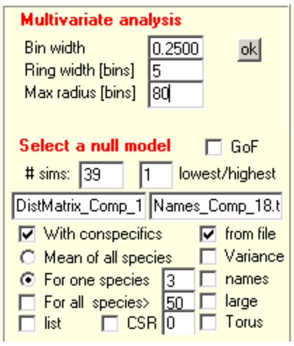
DataType3_Com18_8.phy
DataType3_Com18_9.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy

Which method will you use?
● non-cumulative
○ cumulative

Data MCFunction
● List with coordinates, no grid

The **individual species area relationship** and other individual analyses can be accessed with the following sequence of actions:

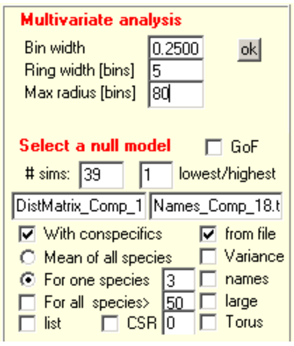
1. Select “**Phylogenetic analysis**” in window What do you want to do?
2. Highlight data file you want to analyze in Input data **(**DataType3_Com18.phy**)**
3. Click “**List with coordinates, no grid**” in Data MCFunction
4. Provide in the window Multivariate analysis the bin width in data units (here use 0.25 to have a better spatial resolution since the competition effect occurs at distances below 5m), an appropriate ring width (here 5 bins = 1.25m), and a maximal distance *r* of the analysis (here 80bin = 20m). However, you can also use a ring width of one and increase the ring width later using the **Replicates** option.
5. Write the names of the distance matrix (DistMatrix_Comp_18.txt) and the species list Names_Comp_18.txt) into the boxes.
6. Click “**calculate simulation envelopes**” to provide settings of null model
7. To run the analysis for individual focal species (here species number 3) select the option “**For one species**” and provide the species number (3 in the example).
8. You can include or exclude the focal species with the check box **With conspecifics.** For the phylogenetic Simpson index use option “With conspecifics”, for the phylogenetic mark correlation function disable this option.



Multivariate analysis
Bin width 0.2500
Ring width [bins] 5
Max radius [bins] 80

Select a null model
□ GoF
# sims: 39
1 lowest/highest
DistMatrix_Comp_1
Names_Comp_18.t
☑ With conspecifics
☑ from file
○ Mean of all species
□ Variance
● For one species 3
□ names
□ For all species> 50
□ large
□ list
□ CSR 0
□ Torus

9. Select the number of simulations of the null model (39) and the rule for the simulation envelopes (1). Usually 199 (or more) realizations of the null model are recommended.

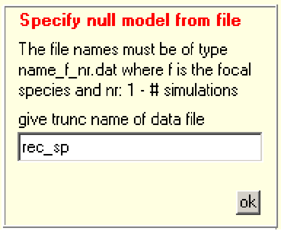


Multivariate analysis
Bin width 0.2500 ok
Ring width [bins] 5
Max radius [bins] 80

Select a null model □ GoF
sims: 39 1 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
☑ With conspecifics ☑ from file
○ Mean of all species □ Variance
● For one species 3 □ names
□ For all species> 50 □ large
□ list □ CSR 0 □ Torus

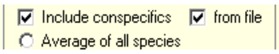
10. To read for the individual analysis the files with the null model locations of the focal species (here number 3) select "**from file**"

After clicking "**from file**" a small window appears where you have to write the trunk name of the null model files. In our example the full names of the null model files are rec_sp3_1.dat, rec_sp3_2.dat, ..., rec_sp3_39.dat where the number of the simulations of the null model run from 1 to 39. *Programita* adds the number of the focal species (3), therefore the **trunk name in the example is "rec_sp"**.



Specify null model from file
The file names must be of type name_f_nr.dat where f is the focal species and nr: 1 - # simulations
give trunc name of data file
rec_sp
ok

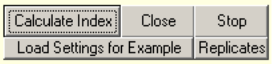
The advantage of this name convention is that you only need to change the number of the focal species in the small text box ● For one species 3 and *Programita* automatically assembles the correct names of the null model files. However, the null model files of the focal species must exist! For example, if the focal species has number 1, the null model files will be rec_sp1_1.dat, rec_sp1_2.dat, ...,

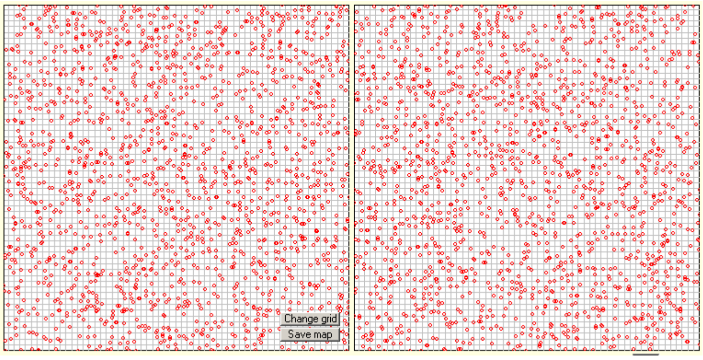


☑ Include conspecifics ☑ from file
○ Average of all species

Finally click the small ok button.

11. Clicking "**Calculate index**" to run the simulations of the null model. *Programita* shows the locations of the focal species (left) and that of the null model patterns of the focal species (right):



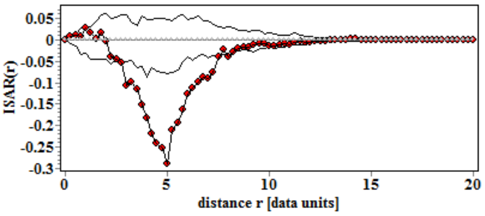


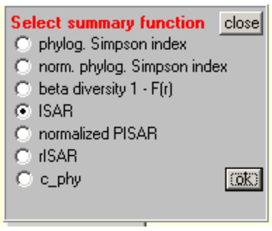
The example data is one of the null communities of Shen et al. (2013) with intra- and interspecific competition among individuals separated by less than 5m. The dissimilarity matrix "DistMatrix_Comp_18.txt" determines the strength of competition between two species.

The null model patterns of the focal species were created by using non-parametric techniques of pattern reconstruction (Wiegand et al. 2013) to exactly conserve the observed characteristics of the patterning of the focal species. This is required to yield null model patterns that are independent from the observed pattern of all other species.

12. You can select with the window Select one test function among different test functions.

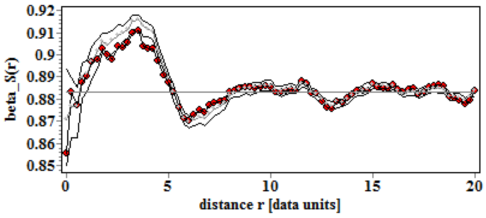
- Select "ISAR" to see results of the **individual species area relationship** *ISAR*$_f$(*r*) for focal species 3:

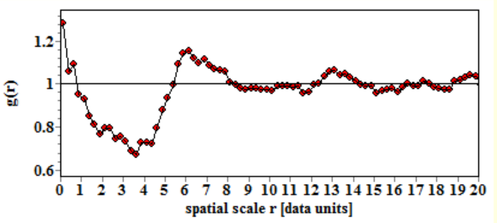



Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy
ok

To visualize departures from the null model better, enable the check box ☑ subtract exp below the results graphs to shows the observed (and pointwise simulation envelopes) minus the expected *ISAR* function. The result indicate that the focal species 3 is at distances between 3 and 8m surrounded by fewer species than expected by the independence null model. Note that the negative peak occurs exactly at distance 5m which is the maximal range of competition.

- Select "beta diversity 1 - F(r)" to obtain the result of the **individual and spatially-explicit Simpson index** $\beta_{f,\mathrm{S}}(r)$ (left)

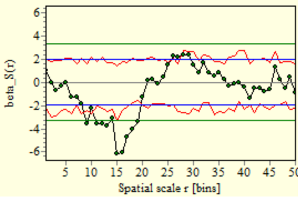




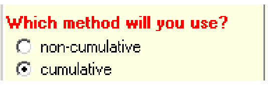
The individual Simpson index $\beta_{f,\mathrm{S}}(r) = 1 - F_\mathrm{f}(r)$ is an inverse measure of the non-cumulative local dominance of the focal species (the local dominance $F_f(r)$ yields the probability that a randomly selected individual of the community that is distance *r* away from an individual of the focal species is conspecific).

The result (left) shows that the competition caused a particular structure in distance decay that however was mostly explained by the univariate pattern of the focal species 3 (right).

The student transformed simulation envelopes (you can access by clicking "**GoF**" and then selection "**student**") show that there is a negative departure at distances of approximately 4 to 5m. Thus, the focal species is not independently placed from the other species. This is because of the negative interspecific species interactions of the Strauss process. There is weak signal of collective negative interactions at these distances.



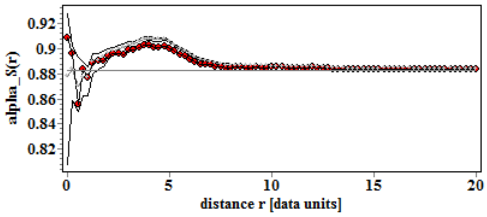
- To obtain a clearer signal of negative interspecific interactions select "**cumulative**" in Which method will you use and run the analysis again to obtain the corresponding cumulative measure of (inverse) local dominance $\alpha_{f,\mathrm{S}}(r)$



Which method will you use?

○ non-cumulative

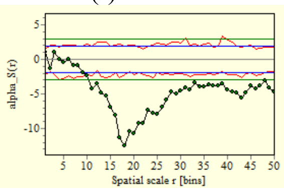
● cumulative



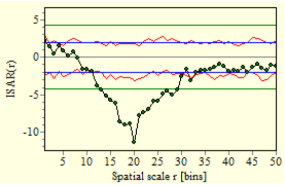
(the cumulative Simpson index $\alpha_{f,\mathrm{S}}(r)$ yields the probability that a randomly selected individual of the community that is located within *r* away from the typical individual of the focal species is heterospecific).

The student transformed simulation envelopes (you can access by clicking "GoF" and then selection "student") show that there is a negative departure at distances larger than 3m. This result is very similar to that of the *ISAR* function, although the cumulative Simpson index $\alpha_{f,\mathrm{S}}(r)$ approaches the simulation envelopes at larger distances (this is clear because the *ISAR* looks only up to the nearest neighbor whereas the Simpson index quantifies also larger scale structures).

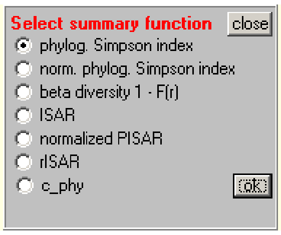
cum 1-F(r):



ISAR:



- Select again "non-cumulative", run the analysis and select "phylog Simpson index" to obtain the **individual phylogenetic Simpson index** $\beta_{f,phyf}(r)$:



Select summary function

● phylog. Simpson index

○ norm. phylog. Simpson index

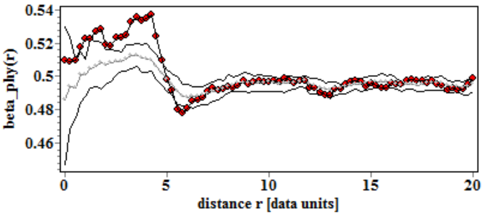
○ beta diversity 1 - F(r)

○ ISAR

○ normalized PISAR

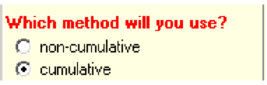
○ rISAR

○ c_phy



The results how that, as expected by generation of the data, the focal species is surrounded at distances below 5m by more dissimilar species. Note the clear and correct indication of the scale effects at 5m.

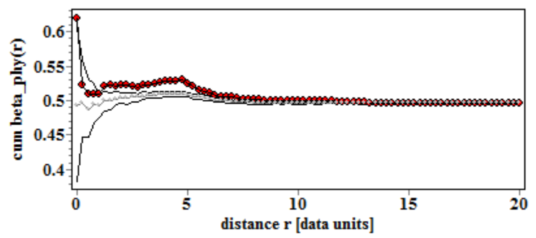
- Select “**cumulative**” in Which method will you use to obtain the corresponding **cumulative phylogenetic Simpson index** $\alpha_{\mathrm{phy,f}}(r)$

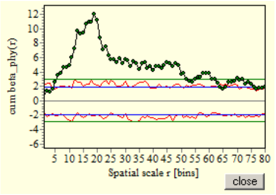


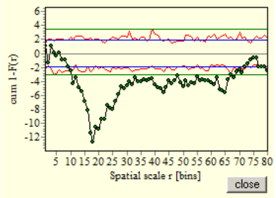
Which method will you use?

○ non-cumulative

● cumulative

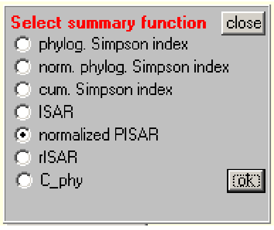






The result shows that the focal species is surrounded up to distances of some 15m by phylogenetically more dissimilar species. However, this result is mostly driven by the underlying structures in local species richness as shown by comparison with the cumulative Simpson index.

- Select “normalized PISAR” to obtain **the normalized *PISAR*** function:



Select summary function

○ phylog. Simpson index

○ norm. phylog. Simpson index

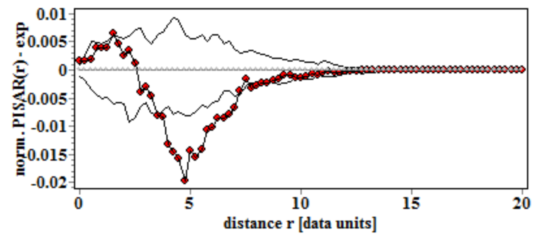
○ cum. Simpson index

○ ISAR

● normalized PISAR

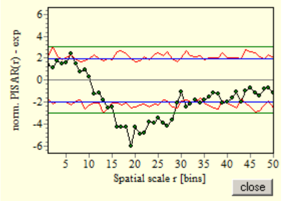
○ rISAR

○ C_phy

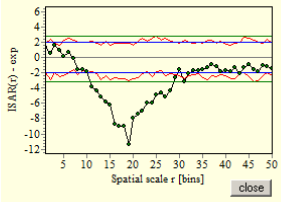


The result is similar to that of the average ISAR, but departures from the null community are somewhat weaker. The average PISAR is therefore driven to a large extent by the underlying species patterns which are captured by the average ISAR. Use the **GoF** test with the **student** transformation to compare the departures with that of the ISAR.

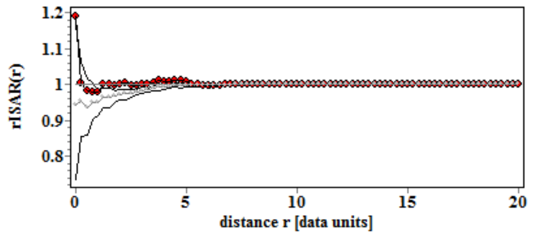
PISAR



ISAR

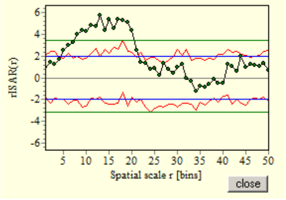


- Select rISAR to obtain the ***rISAR*** function:

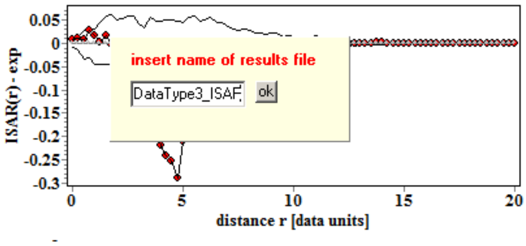


The *rISAR* indicates that the typical individual of the focal species is surrounded up to neighborhoods of some 4m by more dissimilar species than expected by independence. Phylogenetic evenness increases up to distances of 4m (= 16 bins of 0.25m) (due to the competition of more similar species) and then declines smoothly (because it is a cumulative index). Thus, phylogenetic diversity increases quicker than species diversity. This was expected because a species was more likely to be surrounded by more dissimilar species.

rISAR



13. After the analysis is finished save results with the "**Save results**" option



To save results provide name. The results are saved as file name_phy.res and mcf_name_phy.rep.

14. The *.res file contains all your settings and can be used to load all settings to repeat the analysis (using option "**Load Settings for Example**".

The *.rep file contains the detailed results of your analysis that allows you to reassemble them. **However, you can only use the full features of this option if the ring width of the analysis was one**.

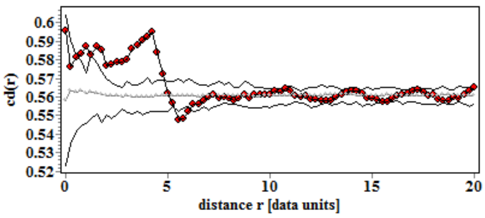
8.3.3 Individual phylogenetic mark correlation functions

15. You can access the individual phylogenetic mark correlation function instead of the phylogenetic Simpson index. The difference is that you include or not conspecifics. To access the phylogenetic mark correlation function **disable** the option "With conspecifics". ☐ Include conspecifics

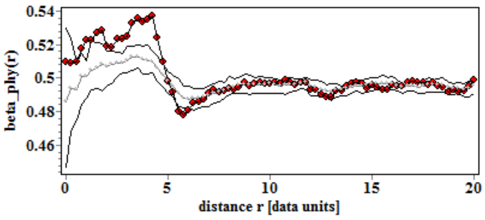
16. After clicking "**Calculate index**" and run the simulations of the null model you can select with the window Select one test function among different test functions:

- Select "non-norm. phylogenetic mcf" to obtain the **individual non-normalized phylogenetic mark correlation function** $c_{d,f}(r)$ of focal species 3.

non-normalized individual phylogenetic mark correlation function



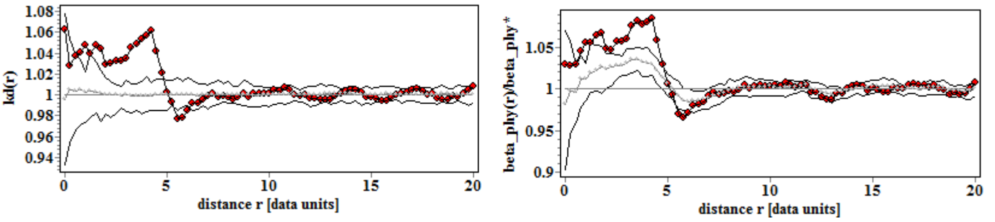
non-normalized individual phylogenetic Simpson index



Comparison with the phylogenetic Simpson index shows that the effect of pure species placement (captured by the Simpson index) is factored out and that the expectation of the null model is a constant. The focal species 3 is, as expected, surrounded at distances between 1 to 5m by phylogenetically more dissimilar species.

- Select “normalized phylogenetic mcf” to obtain the results of the **individual normalized phylogenetic mark correlation function** $k_{d,f}(r)$:

normalized individual phylogenetic mark correlation function   normalized individual phylogenetic Simpson index



The results of the normalized individual phylogenetic mark correlation function $k_{d,f}(r)$ show that the expectation of the null model yields indeed as expected a value of one.

- If you select “beta diversity 1 – F(r)” the beta diversity is shown as before.
- The results of the three summary functions “ISAR”, “PISAR” and “rISAR” that include conspecifics are very similar to the results without conspecifics.

8.3.4 Series of individual analysis for *ISAR* with “from file” option

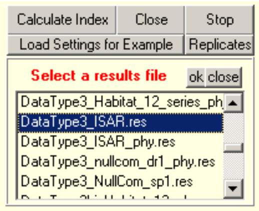
The most interesting feature of the *ISAR* analysis in *Programita* is to conduct a series of analyses for all focal species of a community which are sufficiently abundant (Wiegand et al. 2007). Because the individual analyses need data files of the null model of the focal species you must first generate these files. You also need a list in ASCII format with *.spl extension that contains the number of the focal species to be analyze. The example file is ISAR_spList.spl and it tells *Programita* to analyze focal species 3, 4, and 5:

```
3
4
5
```

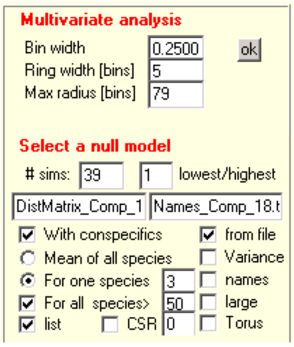
There are two possible ways to code the null model data files for the focal species in the series of analysis procedure. Based on the example *DataType3_ISAR.res*:

- **using the species number**. In this case the first null model files of focal species 3, 4, and 5 are rec_sp3_1.dat, rec_sp4_1.dat, rec_sp5_1.dat, respectively.
- **using the species acronym**. In this case the first null model of focal species 3, 4, and 5 is rec_SPECI3_1.dat, rec_SPECI4_1.dat, rec_SPECI5_1.dat, respectively. The species list is the same, but *Programita* automatically assembles the file name by using the acronyms in the list of species numbers and acronyms (sim10.txt in the example).

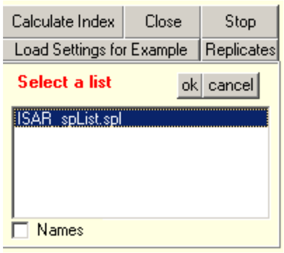
Method using species numbers


Calculate Index
Close
Stop
Load Settings for Example
Replicates
Select a results file
ok close
DataType3_Habitat_12_series_ph
DataType3_ISAR.res
DataType3_ISAR_phy.res
DataType3_nullcom_dr1_phy.res
DataType3_NullCom_sp1.res

1. Load settings from file DataType3_ISAR.res using the “**Load Settings for Example**” option.
2. *Programita* now loads all settings from this analysis. Run the example analysis for focal species 3.



Bin width 0.2500
Ring width [bins] 5
Max radius [bins] 79
# sims: 39 1 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
☑ With conspecifics ☑ from file
○ Mean of all species □ Variance
● For one species 3 □ names
☑ For all species> 50 □ large
☑ list □ CSR 0 □ Torus



ISAR_spList.spl
□ Names

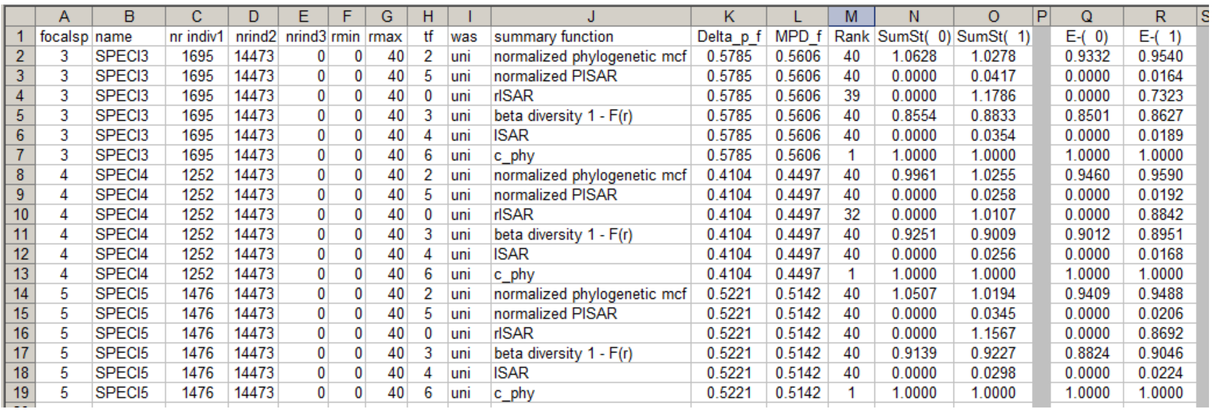
3. Enable the checkbox "**For all species**" and enable the check box "**list**". The window Select a list appears, select the list "ISAR_spList.spl" that contains the 3 focal species 3, 4, 5, and click the small "ok" button

4. Select a summary function in window Select one test function, the *.res results file will use this test function.

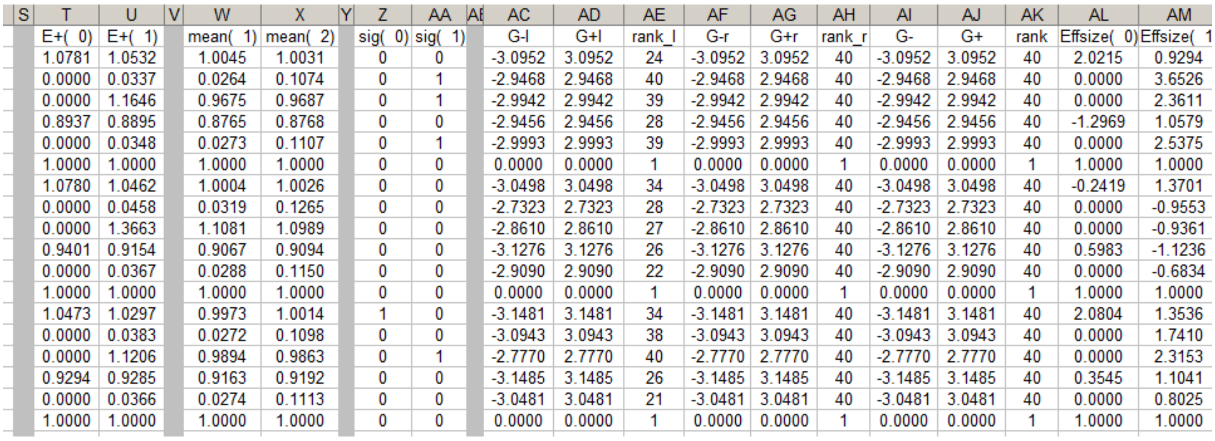
5. Click "**Calculate Index**" and *Programita* runs the individual analyses of all focal species in the list ISAR_spList.spl. To speed up to estimation, disable the options that plot the focal pattern after each simulation of the null model.

6. *Programita* generates results files "name_fsp_nr_phy.res" and if you check the checkbox "large", it saves also the *.rep files "mcf_name_fsp_nr_phy.rep" where "name" is the file name (here "DataType3_Com18") and "nr" the number of the focal species.

7. *Programita* also outputs a file with a results summary named "name.txt" where the name is the name of the data file (here "DataType3_Com18_series.txt"):



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	focalsp	name	nr indiv1	nrind2	nrind3	rmin	rmax	tf	was	summary function	Delta_p_f	MPD_f	Rank	SumSt(0)	SumSt(1)		E-(0)	E-(1)	
2	3	SPECI3	1695	14473	0	0	40	2	uni	normalized phylogenetic mcf	0.5785	0.5606	40	1.0628	1.0278		0.9332	0.9540	
3	3	SPECI3	1695	14473	0	0	40	5	uni	normalized PISAR	0.5785	0.5606	40	0.0000	0.0417		0.0000	0.0164	
4	3	SPECI3	1695	14473	0	0	40	0	uni	rISAR	0.5785	0.5606	39	0.0000	1.1786		0.0000	0.7323	
5	3	SPECI3	1695	14473	0	0	40	3	uni	beta diversity 1 - F(r)	0.5785	0.5606	40	0.8554	0.8833		0.8501	0.8627	
6	3	SPECI3	1695	14473	0	0	40	4	uni	ISAR	0.5785	0.5606	40	0.0000	0.0354		0.0000	0.0189	
7	3	SPECI3	1695	14473	0	0	40	6	uni	c_phy	0.5785	0.5606	1	1.0000	1.0000		1.0000	1.0000	
8	4	SPECI4	1252	14473	0	0	40	2	uni	normalized phylogenetic mcf	0.4104	0.4497	40	0.9961	1.0255		0.9460	0.9590	
9	4	SPECI4	1252	14473	0	0	40	5	uni	normalized PISAR	0.4104	0.4497	40	0.0000	0.0258		0.0000	0.0192	
10	4	SPECI4	1252	14473	0	0	40	0	uni	rISAR	0.4104	0.4497	32	0.0000	1.0107		0.0000	0.8842	
11	4	SPECI4	1252	14473	0	0	40	3	uni	beta diversity 1 - F(r)	0.4104	0.4497	40	0.9251	0.9009		0.9012	0.8951	
12	4	SPECI4	1252	14473	0	0	40	4	uni	ISAR	0.4104	0.4497	40	0.0000	0.0256		0.0000	0.0168	
13	4	SPECI4	1252	14473	0	0	40	6	uni	c_phy	0.4104	0.4497	1	1.0000	1.0000		1.0000	1.0000	
14	5	SPECI5	1476	14473	0	0	40	2	uni	normalized phylogenetic mcf	0.5221	0.5142	40	1.0507	1.0194		0.9409	0.9488	
15	5	SPECI5	1476	14473	0	0	40	5	uni	normalized PISAR	0.5221	0.5142	40	0.0000	0.0345		0.0000	0.0206	
16	5	SPECI5	1476	14473	0	0	40	0	uni	rISAR	0.5221	0.5142	40	0.0000	1.1567		0.0000	0.8692	
17	5	SPECI5	1476	14473	0	0	40	3	uni	beta diversity 1 - F(r)	0.5221	0.5142	40	0.9139	0.9227		0.8824	0.9046	
18	5	SPECI5	1476	14473	0	0	40	4	uni	ISAR	0.5221	0.5142	40	0.0000	0.0298		0.0000	0.0224	
19	5	SPECI5	1476	14473	0	0	40	6	uni	c_phy	0.5221	0.5142	1	1.0000	1.0000		1.0000	1.0000	



S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
	E+(0)	E+(1)		mean(1)	mean(2)		sig(0)	sig(1)		G-l	G+l	rank_l	G-r	G+r	rank_r	G-	G+	rank	Effsize(0)	Effsize(1
	1.0781	1.0532		1.0045	1.0031		0	0		-3.0952	3.0952	24	-3.0952	3.0952	40	-3.0952	3.0952	40	2.0215	0.9294
	0.0000	0.0337		0.0264	0.1074		0	1		-2.9468	2.9468	40	-2.9468	2.9468	40	-2.9468	2.9468	40	0.0000	3.6526
	0.0000	1.1646		0.9675	0.9687		0	1		-2.9942	2.9942	39	-2.9942	2.9942	40	-2.9942	2.9942	40	0.0000	2.3611
	0.8937	0.8895		0.8765	0.8768		0	0		-2.9456	2.9456	28	-2.9456	2.9456	40	-2.9456	2.9456	40	-1.2969	1.0579
	0.0000	0.0348		0.0273	0.1107		0	1		-2.9993	2.9993	39	-2.9993	2.9993	40	-2.9993	2.9993	40	0.0000	2.5375
	1.0000	1.0000		1.0000	1.0000		0	0		0.0000	0.0000	1	0.0000	0.0000	1	0.0000	0.0000	1	1.0000	1.0000
	1.0780	1.0462		1.0004	1.0026		0	0		-3.0498	3.0498	34	-3.0498	3.0498	40	-3.0498	3.0498	40	-0.2419	1.3701
	0.0000	0.0458		0.0319	0.1265		0	0		-2.7323	2.7323	28	-2.7323	2.7323	40	-2.7323	2.7323	40	0.0000	-0.9553
	0.0000	1.3663		1.1081	1.0989		0	0		-2.8610	2.8610	27	-2.8610	2.8610	40	-2.8610	2.8610	40	0.0000	-0.9361
	0.9401	0.9154		0.9067	0.9094		0	0		-3.1276	3.1276	26	-3.1276	3.1276	40	-3.1276	3.1276	40	0.5983	-1.1236
	0.0000	0.0367		0.0288	0.1150		0	0		-2.9090	2.9090	22	-2.9090	2.9090	40	-2.9090	2.9090	40	0.0000	-0.6834
	1.0000	1.0000		1.0000	1.0000		0	0		0.0000	0.0000	1	0.0000	0.0000	1	0.0000	0.0000	1	1.0000	1.0000
	1.0473	1.0297		0.9973	1.0014		1	0		-3.1481	3.1481	34	-3.1481	3.1481	40	-3.1481	3.1481	40	2.0804	1.3536
	0.0000	0.0383		0.0272	0.1098		0	0		-3.0943	3.0943	38	-3.0943	3.0943	40	-3.0943	3.0943	40	0.0000	1.7410
	0.0000	1.1206		0.9894	0.9863		0	1		-2.7770	2.7770	40	-2.7770	2.7770	40	-2.7770	2.7770	40	0.0000	2.3153
	0.9294	0.9285		0.9163	0.9192		0	0		-3.1485	3.1485	26	-3.1485	3.1485	40	-3.1485	3.1485	40	0.3545	1.1041
	0.0000	0.0366		0.0274	0.1113		0	0		-3.0481	3.0481	21	-3.0481	3.0481	40	-3.0481	3.0481	40	0.0000	0.8025
	1.0000	1.0000		1.0000	1.0000		0	0		0.0000	0.0000	1	0.0000	0.0000	1	0.0000	0.0000	1	1.0000	1.0000

8. The results summary provides information on:

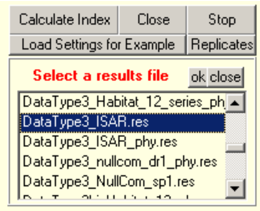
- focalsp: the species number of the focal species
- name: the acronym of the focal species
- nr indiv1: number of individuals of the focal species
- nrind2: number of individuals of the first multivariate pattern
- nrind3: number of individuals of the second multivariate pattern (always 0 for analysis with one multivariate pattern)
- rmin: minimal distance for GoF test
- rmax: maximal distance for GoF test
- tf: number of summary function
- was “uni” or “bi” indicates if results are from “univariate” analysis (i.e., one multivariate pattern) or “bivariate” (i.e., two multivariate patterns).
- summary function
- Delta_p_f normalization constant of *rISAR*
- MPD_f normalization constant of $k_{\mathrm{d}}(r)$
- Rank: the rank of the standard GoF test over interval rmin-rmax
- SumSt(r) the value of the summary functions at distance *r*.
- E-(r) the value of lower simulation envelop at distance *r*.
- E+(r) the value of upper simulation envelop at distance *r*.
- mean(r) the expected value of the summary functions at distance *r*.
- sig(r))) 0: inside simulation envelopes, -1: below simulation envelopes and 1: above simulation envelops.

Additionally information on the global envelope test:

- G-l lower global envelope over interval 1 to rmax/2
- G+l upper global envelope over interval 1 to rmax/2
- rank_l rank of global envelope test over interval 1 to rmax/2
- G-r lower global envelope over interval rmax/2 to radmax
- G+r upper global envelope over interval rmax/2 to radmax
- rank_r rank of global envelope test over interval rmax/2 to radmax
- G- lower global envelope over interval 1 to rmax
- G+ upper global envelope over interval 1 to rmax
- rank rank of global envelope test over interval 1 to rmax
- Effsize(r) effect size for distance r

Method using species acronyms

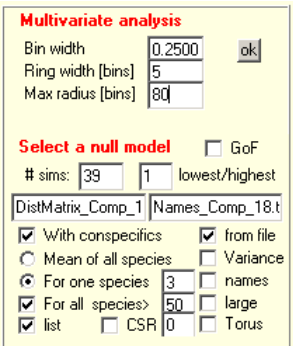
1. Load settings from file DataType3_ISAR.res using the "**Load Settings for Example**" option.



DataType3_Habitat_12_series_ph
DataType3_ISAR.res
DataType3_ISAR_phy.res
DataType3_nullcom_dr1_phy.res
DataType3_NullCom_sp1.res

2. *Programita* now loads all settings from this analysis. Run the example analysis for focal species 3.

3. Enable the checkbox "**For all species**" and enable the check box "**list**". The window Select a list appears, select the list "ISAR_spList.spl" that contains the 3 focal species 3, 4, 5.

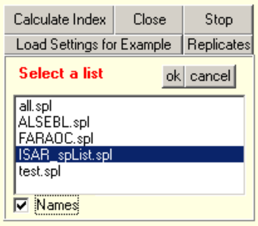


Bin width 0.2500
Ring width [bins] 5
Max radius [bins] 80

□ GoF
# sims: 39 1 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
☑ With conspecifics ☑ from file
○ Mean of all species □ Variance
● For one species 3 □ names
☑ For all species> 50 □ large
☑ list □ CSR 0 □ Torus

4. Click also "**Names**" to tell *Programita* that the null model files are composed by the species acronym. Finally, click the small "**ok**" button

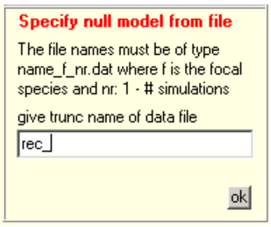
5. Click two times the box "**from file**" to open the window "Specify null model from file". Because the null model files are named rec_SPECI3_1.dat, rec_SPECI4_1.dat, etc, the trunk name is "rec_".



all.spl
ALSEBL.spl
FARAOC.spl
ISAR_spList.spl
test.spl
☑ Names

6. Select a summary function in window Select one test function, the *.res results file will use this test function. Finally, click the small "**ok**" button

7. Click "**Calculate Index**" and *Programita* runs the individual analyses of all focal species in the list ISAR_spList.spl. To speed up to estimation, disable the options that plot the focal pattern after each simulation of the null model.



The file names must be of type name_f_nr.dat where f is the focal species and nr: 1 - # simulations

give trunc name of data file

rec_

8. *Programita* generates results files "name_fsp_nr_phy.res" and "mcf_name_fsp_nr_phy.rep" where "name" is the file name (here "DataType3_Com18") and "nr" the number of the focal species.

9. *Programita* outputs, as above, a file with a results summary named "name.txt" where the name is the name of the data file (here "DataType3_Com18_series.txt").

8.3.5 Individual analyses and *Programita* null models

I implemented several simple null models in *Programita* that allow you for quick checks of spatial structure in individual analysis of multivariate data sets. The null models are based on the *Programita* null communities presented above. However, while the null communities randomize the locations of all individuals in the community and the analysis quantifies the spatial structure of the null communities, the **individual analyses randomize only the locations of the focal species and keep the individuals of all other species at their observed locations**. Thus, basically, they compare different properties of the biotic neighborhood of the individuals of the focal species with that of random locations in the plot (i.e., the null model locations). **This is a fundamental difference between the community level and the individual level analysis**. The null models implemented in *Programita* include:

- CSR (random placement)
- a non-parametric heterogeneous Poisson process (habitat filtering)
- a global toroidal shift (dispersal limitation)
- a local toroidal shift (combined habitat and dispersal limitation)
- local and global random labeling

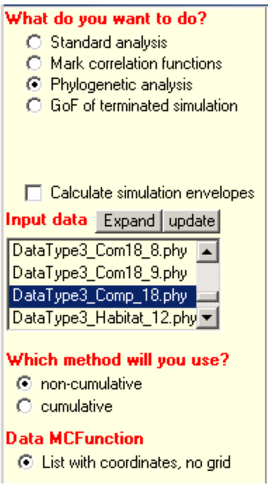
All null models maintain the observed abundances of the focal species in the observation window. The null models can be accessed in the "multivariate analysis" null model window via .

○ CSR, R= 20 ● Torus ○ RL

However, more refined null communities such as that presented in Wang et al. (2016) need to be generated outside of *Programita* and uploaded with the "from file" option.

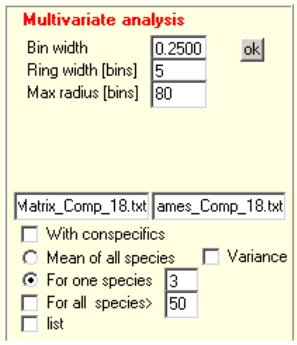
8.3.6 Individual analysis and toroidal shift null model

The multivariate analysis based on the toroidal null communities can be accessed with the following sequence of actions:



What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation
□ Calculate simulation envelopes
Input data Expand update
DataType3_Com18_8.phy
DataType3_Com18_9.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy
Which method will you use?
● non-cumulative
○ cumulative
Data MCFunction
● List with coordinates, no grid

1. Select "**Phylogenetic analysis**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data (DataType3_Com18.phy)
3. Click "**List with coordinates, no grid**" in Data MCFunction



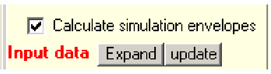
Multivariate analysis
Bin width 0.2500 ok
Ring width [bins] 5
Max radius [bins] 80
Matrix_Comp_18.txt ames_Comp_18.txt
□ With conspecifics
○ Mean of all species □ Variance
● For one species 3
□ For all species> 50
□ list

4. Provide in the window Multivariate analysis the bin width in data units (here use 0.25 to have a better spatial resolution since the competition effect occurs at distances below 5m), an appropriate ring width (here 5 bins = 1.25m), and a maximal distance *r* of the analysis (here 80bin = 20m). However, you can also use a ring width of one and increase the ring width later using the **Replicates** option.
5. Write the names of the distance matrix (DistMatrix_Comp_18.txt) and the species list (Names_Comp_18.txt) into the boxes.
6. For the standard individual analysis select "**For one species**" and "**3**" for the number of the focal species.

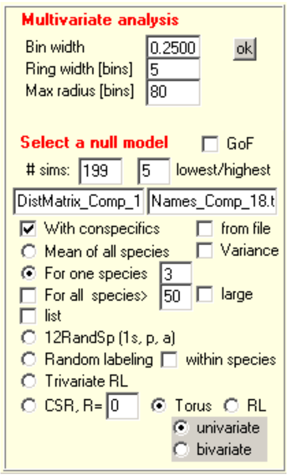
7. You can include or exclude the focal species with the check box **With conspecifics.** For the phylogenetic Simpson index use option "With conspecifics", for the phylogenetic mark correlation function disable this option.

8. Enable the checkbox "Calculate simulation envelopes" and select the "Torus" null model "Multivariate analysis".

9. Select the number of simulations of the null model (199) and the rule for the simulation envelopes (5). Usually 199 (or more) realizations of the null model are recommended.



☑ Calculate simulation envelopes
Input data Expand update

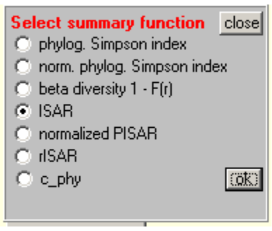


Multivariate analysis
Bin width 0.2500 ok
Ring width [bins] 5
Max radius [bins] 80

Select a null model □ GoF
# sims: 199 5 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
☑ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 3
□ For all species> 50 □ large
□ list
○ 12RandSp (1s, p, a)
○ Random labeling □ within species
○ Trivariate RL
○ CSR, R= 0 ● Torus ○ RL
● univariate
○ bivariate

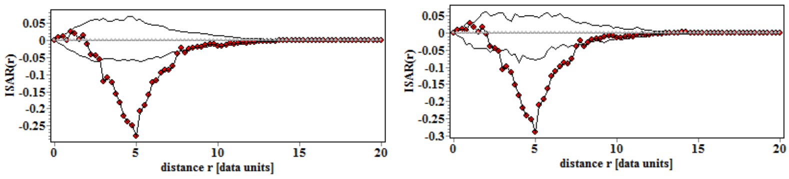
10. Click "**Calculate index**" and *Programita* estimates the different indices of the data.

11. Select "ISAR" to see results of the **individual species area relationship** $ISAR_f(r)$ for focal species 3:



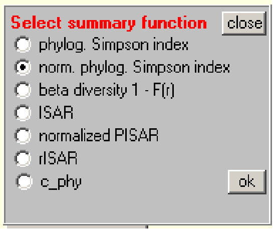
Select summary function close
○ phylog. Simpson index
○ norm. phylog. Simpson index
○ beta diversity 1 - F(r)
● ISAR
○ normalized PISAR
○ rISAR
○ c_phy
ok

toroidal shift   pattern reconstruction

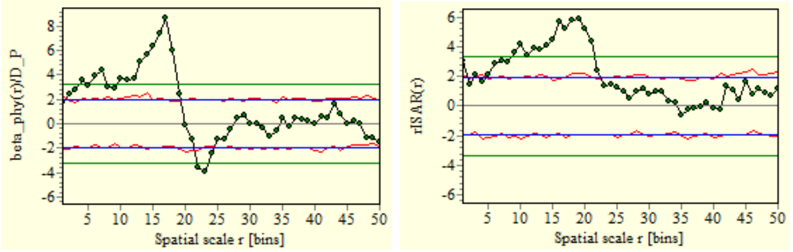


Comparison with the results of the pattern reconstruction null model show very little differences. This shows that the toroidal shift effectively removes small-scale associations.

12. The phylogenetic Simpson index and the *rISAR* function show that the focal species 3 is up to 5m (20 bins of 0.25m) surrounded by more dissimilar species as expected by random locations in the plot. The non-cumulative phylogenetic Simpson index shows also the jump to more similar species just outside the 5m zone of competition.

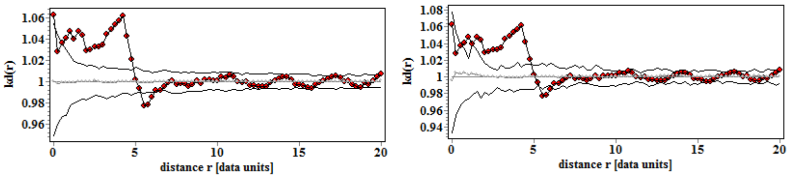


Select summary function close
○ phylog. Simpson index
● norm. phylog. Simpson index
○ beta diversity 1 - F(r)
○ ISAR
○ normalized PISAR
○ rISAR
○ c_phy
ok



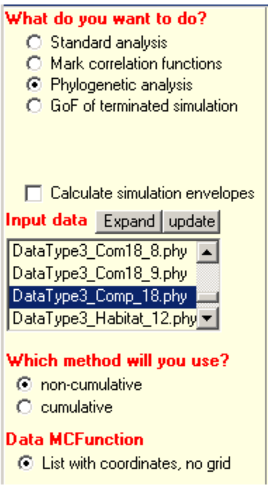
13. Disable "With conspecifics", repeat the analysis, and select "beta diversity 1 - F(r)" to obtain the result of the individual **phylogenetic mark correlation function** $k_{f,d}(r)$ that is largely unaffected by the pattern of the focal species:

toroidal shift   pattern reconstruction

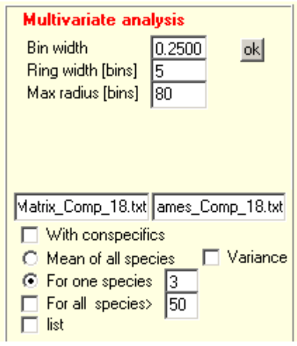


8.3.7 Individual analysis and CSR null model

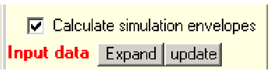
The multivariate analysis based on the toroidal null communities can be accessed with the following sequence of actions:


What do you want to do?
○ Standard analysis
○ Mark correlation functions
● Phylogenetic analysis
○ GoF of terminated simulation
□ Calculate simulation envelopes
Input data Expand update
DataType3_Com18_8.phy
DataType3_Com18_9.phy
DataType3_Comp_18.phy
DataType3_Habitat_12.phy
Which method will you use?
● non-cumulative
○ cumulative
Data MCFunction
● List with coordinates, no grid

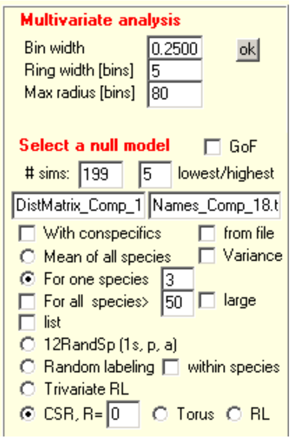
1. Select "**Phylogenetic analysis**" in window What do you want to do?
2. Highlight data file you want to analyze in Input data (DataType3_Com18.phy)
3. Click "**List with coordinates, no grid**" in Data MCFunction
4. Provide in the window Multivariate analysis the bin width in data units (here use 0.25 to have a better spatial resolution since the competition effect occurs at distances below 5m), an appropriate ring width (here 5 bins = 1.25m), and a maximal distance *r* of the analysis (here 80bin = 20m). However, you can also use a ring width of one and increase the ring width later using the **Replicates** option.
5. Write the names of the distance matrix (DistMatrix_Comp_18.txt) and the species list (Names_Comp_18.txt) into the boxes.
6. For the standard individual analysis select "**For one species**" and "**3**" for the number of the focal species.


Multivariate analysis
Bin width 0.2500 ok
Ring width [bins] 5
Max radius [bins] 80
Matrix_Comp_18.txt ames_Comp_18.txt
□ With conspecifics
○ Mean of all species □ Variance
● For one species 3
□ For all species> 50
□ list

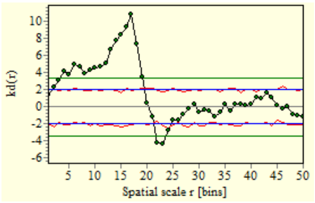
7. You can include or exclude the focal species with the check box **With conspecifics.** For the phylogenetic Simpson index use option "With conspecifics", for the phylogenetic mark correlation function disable this option.
8. Enable the checkbox "Calculate simulation envelopes" and select the "CSR, R=" null model in the "Multivariate analysis" window with R= 0.

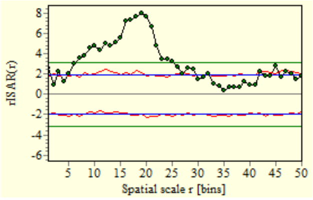

☑ Calculate simulation envelopes
Input data Expand update

9. Select the number of simulations of the null model (199) and the rule for the simulation envelopes (5). Usually 199 (or more) realizations of the null model are recommended.
10. Click "**Calculate index**" and *Programita* estimates the different summary functions of the data and the null model.


Multivariate analysis
Bin width 0.2500 ok
Ring width [bins] 5
Max radius [bins] 80
Select a null model □ GoF
# sims: 199 5 lowest/highest
DistMatrix_Comp_1 Names_Comp_18.t
□ With conspecifics □ from file
○ Mean of all species □ Variance
● For one species 3
□ For all species> 50 □ large
□ list
○ 12RandSp (1s, p, a)
○ Random labeling □ within species
○ Trivariate RL
● CSR, R= 0 ○ Torus ○ RL

14. The phylogenetic mark correlation function and the *rISAR* function show that the focal species 3 is up to 5m (20 bins of 0.25m) surrounded by more dissimilar species as expected by random locations in the plot. The non-cumulative phylogenetic Simpson index shows also the jump to more similar species just outside the 5m zone of competition.





8.4 Two multivariate patterns using a dissimilarity matrix

Programita allows you also to analyze spatial structures among individuals of two types of points of the same community. For example, you can analyze the phylogenetic diversity of small trees around large trees by using a “bivariate” phylogenetic Simpson index. In this case, the “**focal individuals**” belong to the first pattern of large trees and the “**counted individuals**” belong the second pattern of small trees. For example, the estimator of the “bivariate” phylogenetic Simpson index is formally the same as for the “univariate” phylogenetic Simpson index:

$$\beta_{phy}(r) = \sum_{f=1}^{S}\sum_{m=1}^{S} p_{fm}(r)\delta_{fm}^{P} \ ,$$

but now the mark connection functions $p_{fm}(r)$ yield the probability that, when randomly selecting a large tree and a small tree distance r apart, the large tree is of type f and the small tree of type m.

Based on the same principle, all summary functions, including those based on an individual focal species, listed in the overview table (see below) can be applied in a “bivariate” manner. The focal individuals are always taken from the first multivariate pattern (e.g., large trees) and the counted individuals are always taken from the second multivariate pattern (e.g., small trees). Note that some of the normalization constant have also be re-interpreted accordingly. For example, in the estimation of the indices D^{P} and D_f^{P}

$$D^{\mathrm{P}} = \sum_{i=1}^{S}\sum_{j=1}^{S} \delta_{ij}^{P} f_i f_j \quad \text{and} \quad D_f^{\mathrm{P}} = \sum_{j=1}^{S} \delta_{fj}^{P} f_j \ ,$$

the f_i refers to the relative abundance of species i within the first multivariate pattern whereas the f_j refers to the relative abundance of species j within the second multivariate pattern. The same is true for of the indices S^{P}:

$$S^{\mathrm{P}} = \sum_{i=1}^{S} f_i \sum_{i=1}^{S} \delta_{ij}^{P} \ .$$

classifier				non-spatial metrics			spatial metrics			spatial
				A	B	C	A	B	C	condition
F1	α	S	community	S^{S}	S^{P}	$\Delta^{\mathrm{P}*} = S^{\mathrm{P}}/S^{\mathrm{S}}$	$\overline{ISAR}(r)$	$\overline{PISAR}(r)$	$\overline{rISAR}(r)$	$D_{fs}(r)$
F2	α	S	focal species	S_f	S_f^{P}	$\Delta_f^{\mathrm{P}} = S_f^{\mathrm{P}}/S_f$	$ISAR_f(r)$	$PISAR_f(r)$	$rISAR_f(r)$	$D_{fs}(r)$
F3	α	D	community	D	D^{P}	$c_{\mathrm{d}} = D^{\mathrm{P}}/D$	$\alpha_{\mathrm{S}}(r)$	$\alpha_{\mathrm{phy}}(r)$	$K_{\mathrm{d}}(r)$	$K_{ij}(r)/K(r)$
F4	α	D	focal species	D_f	D_f^{P}	$c_{f\mathrm{d}} = D_f^{\mathrm{P}}/D_f$	$\alpha_{f,\mathrm{S}}(r)$	$\alpha_{f,\mathrm{phy}}(r)$	$K_{f,\mathrm{d}}(r)$	$K_{ij}(r)/K(r)$
F5*	β	S	community	S^{S}	S^{P}	$\Delta^{\mathrm{P}*} = S^{\mathrm{P}}/S^{\mathrm{S}}$	$\overline{isar}(r)$	$\overline{pisar}(r)$	$\overline{risar}(r)$	$d_{fs}(r)$
F6*	β	S	focal species	S_f	S_f^{P}	$\Delta_f^{\mathrm{P}} = S_f^{\mathrm{P}}/S_f$	$isar_f(r)$	$pisar_f(r)$	$risar_f(r)$	$d_{fs}(r)$
F7	β	D	community	D	D^{P}	$c_{\mathrm{d}} = D^{\mathrm{P}}/D$	$\beta_{\mathrm{S}}(r)$	$\beta_{\mathrm{phy}}(r)$	$k_{\mathrm{d}}(r)$	$g_{ij}(r)/g(r)$
F8	β	D	focal species	D_f	D_f^{P}	$c_{f\mathrm{d}} = D_f^{\mathrm{P}}/D_f$	$\beta_{f,\mathrm{S}}(r)$	$\beta_{f,\mathrm{phy}}(r)$	$k_{f,\mathrm{d}}(r)$	$g_{ij}(r)/g(r)$

*Metric families F5 and F6 that are based on the non-cumulative probability density function $d_{ij}(r)$ of the distances to the nearest species j neighbor have not been used to date.

8.4.1 Data preparation for analysis of two multivariate pattern

For the multivariate analysis you need three (or four) data sets:

1. a data file with the location and species identity of all individuals in the two multivariate patterns. This is an ASCII file with *.phy extension.
2. a data file with the species acronyms and the species numbers. This is an ASCII file with *.txt extension
3. a data file with dissimilarity matrix. This is an ASCII file with *.txt extension
4. if the null communities or null model files were created outside *Programita*, you need additionally the *.phy null community files (for community level analysis) or the *.dat focal species null model files (for individual analysis e.g., with the *ISAR* family)

1) The data files for "bivariate" average analysis to detect phylogenetic (or functional) spatial structure in the fine-scale placement among individuals of two types are given by two multivariate patterns. For example, the focal pattern could be that of large trees and the second pattern that of small trees. The data files must be an ASCII file with *. phy extension and have the following format (the example data file DataType3bi_cluster10.phy):

```
0  316  0  316  10101
 304.98  203.69   1     1  100
 236.99  311.00   1     2  100
 289.42  171.37   1     3  100
 187.55   11.82   1     4  100
 308.96  138.19   1     5  100
   5.14  210.25   1     6  100
 176.71   76.73   2     1  100
  74.92   29.00   2     2  100
 271.00   39.00   2     3  100
 236.51  164.10   2     4  100
 227.24   31.34   2     5  100
…
```

- the first line gives the dimension of the plot (316 × 316 units) in the example and the total number of points in the list (10,101 in the example)
- the first two columns are the coordinates of the points
- the third column gives the pattern: "1" for the first focal multivariate pattern (e.g., large trees) and "2" for the second multivariate pattern (e.g., small trees)
- the forth column gives the species identifier (being an integer running from 1 to *S*).
- the fifth column is optional and can carry an quantitative mark, however, this mark is not yet used.

2) The file with species numbers and species acronyms (here file Names_random1.txt) is a **tab (or space) delimited** ASCII file with the *.txt extension and the following format:

```
1   SPECI1
2   SPECI2
…
9   SPECI9
10  SPEC10
….
```

where the first column is the species number and the second column an up to 6 letter species acronym.

3) The data file with distance matrix (here Dist_random1.txt) which is a **tab (or space) delimited** ASCII file with the *.txt extension and the following format:

```
SPECI2 SPECI1  15.9161
SPECI3 SPECI1  19.6389
SPECI3 SPECI2  19.6389
SPECI4 SPECI1  19.6389
SPECI4 SPECI2  19.6389
SPECI4 SPECI3  11.8997
SPECI5 SPECI1  19.6389
….
```

where the first two column are the six letter species acronyms of the species pair and the third column is the distance between the two species.

Note that this file must be tab or space delimited and that the species acronyms in the distance matrix and the species list must exactly match.

4a) The null community files

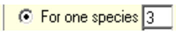
The null community files must have the same format as the *.phy data files. They must also have the same number of individuals and the same species abundance distribution. The name of the data file and the null model files must be the same, except the number at the end:

DataType3_Com18_0.phy (observed data of the community)
DataType3_Com18_1.phy (first null community)
DataType3_Com18_2.phy (second null community)

….
DataType3_Com18_39.phy (last null community).

4b) The null model files for the focal species

The null model files of the focal species must be standard univariate *.dat files with the same number of individuals as the observed focal species. If you use the pattern reconstruction software, the output files are already in the right format. If the original file of the focal species is named SPECI3.dat, the reconstructed files will be named
rec_SPECI3_1.dat
rec_SPECI3_2.dat
rec_SPECI3_3.dat
….
rec_SPECI3_n.dat

were the number of the simulations of the null model run from 1 to n. The advantage of this name convention is that you only need to change the number of the focal species in the small text box 

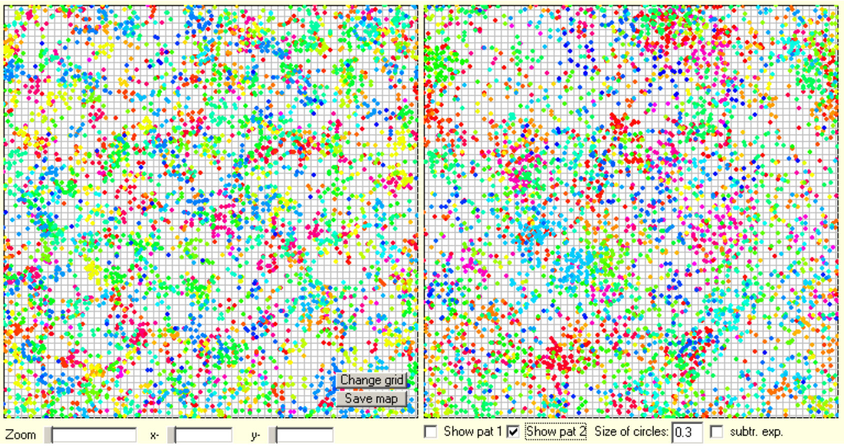
and *Programita* automatically assembles the correct names of the null model files. However, the null model files of the focal species must exist!

8.4.2 Example file for two multivariate patterns

I show an example for the analysis of two multivariate patterns that is based on the data file “DataType3bi_cluster10.phy”. The first multivariate pattern (i.e., large trees) is an example of the “dispersal limitation” communities presented in Wiegand et al. (2017). It mimicked dispersal limitation by using for each species a Thomas cluster process with parameters σ = 5m and ρ = 0.0002/m^2. Species were placed without regard to each other. The dissimilarity matrix and species abundances were taken from a “random community” generated with the R package metricTester presented in Miller et al. (2017). Thus, the spatial pattern of large trees is only governed by dispersal limitation and does not contain phylogenetic spatial structure or structure due to habitat filtering.

The second multivariate pattern (i.e., small trees) was generated by a mixture of a random pattern (10%) and a Gaussian dispersal kernel (90%) around the individuals of the first multivariate pattern with parameter σ = 10m where only 20% of the large trees generated offspring:

first multivariate pattern second multivariate pattern



☐ Show pat 1 ☑ Show pat 2 Size of circles: 0.3 ☐ subtr. exp.

Thus, there is also no spatial phylogenetic structure within the small trees and no spatial phylogenetic structure between small and large trees. However, due to the dispersal kernel, there is a distance decay of similarity between the community of small and large trees.

9 The grid-based standard analysis

Coming soon

10 Using *Programita* in the mode for object of finite size and real shape

Coming soon

11 References

Biganzoli, F., T. Wiegand and W.B. Batista. 2009. Fire-mediated interactions between shrubs in a South American temperate savannah. Oikos 118: 1383-1395

Chanthorn, W., S. Getzin, T. Wiegand, W.Y. Brockelman, and A. Nathalang. Spatial patterns of local species richness reveal importance of frugivores for tropical forest diversity. *Journal of Ecology* DOI: 10.1111/1365-2745.12886

Chave J, Leigh EG. 2002. A spatially explicit neutral model of beta-diversity in tropical forests. *Theor. Popul. Biol.* 62:153–168

Clarke KR, Warwick RM. 1998. A taxonomic distinctness index and its statistical properties. *J. Appl. Ecol.* 35(4):523–31

Condit R, Pitman N, Leigh EG, Chave J, Terborgh J, et al. 2002. Beta diversity in tropical forest trees. *Science* 295:666–69

De la Cruz Rot M. 2006. Introducción al análisis de datos mapeados o algunas de las (muchas) cosas que puedo hacer si tengo coordenadas. Ecosistemas. 2006/3

De la Cruz Rot, M., R. L. Romao, A. Escudero, and F. T. Maestre. 2008. Where do seedlings go? A spatio-temporal analysis of seedling mortality in a semi-arid gypsophyte. Ecography 31: 1–11.

Diggle, P. J. 2003. *Statistical analysis of spatial point patterns*. 2nd ed. Arnold, London

Diggle P.J., V. Gómez-Rubio, P.E. Brown, A.G. Chetwynd, and S. Gooding. 2007. Second-order analysis of inhomogeneous spatial point processes using case-control data. Biometrics 63: 550-557.

Faith DP. 1992. Conservation evaluation and phylogenetic diversity. *Biol. Conserv.* 61:1–10

Fedriani, J, T. Wiegand, G. Calvo, A. Suarez-Esteban, M. Jacome, M. Zywiec, and M. Delibes. 2015. Unravelling conflicting density- and distance-dependent effects on plant reproduction using a spatially explicit approach. *Journal of Ecology*: 103: 1344-1353

Getzin, S. and T. Wiegand. 2014. Stochastically driven adult-recruit associations of tree species on Barro Colorado Island. *Proceedings R. Soc. B* 281: 20140922

Goreaud, F., and R. Pelissier. 2003. Avoiding misinterpretation of biotic interactions with the intertype *K*-12-function: population independence vs. random labelling hypotheses. Journal of Vegetation Science 14:681–692. 2010

Grimm, V. and S.F. Railsback. 2005. *Individual-based Modeling and Ecology*. Princeton University Press

Hardy OJ, Senterre B. 2007.Characterizing the phylogenetic structure of communities by an additive partitioning of phylogenetic diversity. *J. Ecol.* 95: 493–506

Illian, J.B., J. Møller, and R. Waagepetersen. 2009. Hierarchical spatial point process analysis for a plant community with high biodiversity. Environmental and Ecological Statistics 16: 389–405

Jacquemyn, H., P. Endels, O. Honnay, and T. Wiegand. 2010. Evaluating management interventions in small populations of a perennial herb *Primula vulgaris* using spatio-temporal analyses of point patterns. *Journal of Applied Ecology* 47: 431–440

Jacquemyn, H., R. Brys, V.S.F.T. Merckx, M. Waud, B. Lievens, and T. Wiegand. 2014. Co-existing orchid species have distinct mycorrhizal communities and display strong spatial segregation. *New Phytologist* 202: 616–627

Kwan, C.C. 2011. An introduction to shrinkage estimation of the covariance matrix: A pedagogic illustration. Spreadsheets in Education: 4, art. 6.

Loosmore, N. B., and E. D. Ford. 2006. Statistical inference using the *G* or *K* point pattern spatial statistics. Ecology 87:1925–1931.

Martínez, I., T. Wiegand, F. Glez. Taboada, and J. R. Obeso. 2010. Spatial associations among tree species in a temperate forest community in North-western Spain. Forest *Ecology and Management* 260: 456-465

Murphy, S.J., T. Wiegand and L. Comita. 2017. Distance-dependent seedling mortality and long-term spacing dynamics in a neotropical forest community. *Ecology Letters* 20: 1469-1478

Myllymäki, M., T. Mrkvicka, P. Grabarnik, H. Seijo and U. Hahn. 2017. Global envelope tests for spatial processes. Journal of the Royal Statistical Society Series B 79: 381–404.

Myllymäki M., et al. 2015. Deviation test construction and power comparison for marked spatial point patterns. Spatial Statistics 11:19–34.

Parmentier I, Réjou-Méchain M, Chave J, Vleminckx J, Thomas DW, et al. 2014. Prevalence of phylogenetic clustering at multiple scales in an African rain forest tree community. *J. Ecol.* 102:1008–16

Punchi-Manage, R., T. Wiegand, K. Wiegand, S. Getzin, C.V.S. Gunatilleke, and I.A.U. N Gunatilleke. 2015. Neighborhood diversity of large trees shows independent species patterns in a mixed dipterocarp forest in Sri Lanka. *Ecology* 96: 1823–1834

Raventós, J., T. Wiegand, and M. De Luis. 2010. Evidence for the spatial segregation hypothesis: a test with nine-year survivorship data in a Mediterranean fire-prone shrubland show that interspecific and density-dependent spatial interactions dominate. Ecology 91:2110-2120.

Ripley, B.D. 1981. Spatial statistics. John Wiley Sons.

Shao, X., Brown, C., Worthy, S.J., Liu, L., Cao, M., Li, Q., Lin, L., and N.G.Swenson. 2018. Intra-specific relatedness, spatial clustering and reduced demographic performance in tropical rainforest trees. *Ecology Letters*.

Shen, G., T. Wiegand, X. Mi, and F. He. 2013. Quantifying spatial phylogenetic structures of fully mapped plant communities. Methods in Ecology and Evolution 4: 1132-1141

Stoyan, D., and H. Stoyan. 1996. Estimating pair correlation functions of planar cluster processes. Biometrical Journal 38: 259–271.

Velázquez, E., M. Kazmierczak, and T. Wiegand. 2016a. Spatial patterns of sapling mortality in a moist tropical forest: consistency with species-blind density-dependent effects. *Oikos* 125: 872–882

Velázquez, E., I. Martínez, S. Getzin, K.A. Moloney, and T. Wiegand. 2016b. An evaluation of the state of spatial point pattern analysis in ecology. Ecography 39: 1042–1055

Wang, X., T. Wiegand, Z. Hao, B. Li, J. Ye, and J. Zhang. 2010. Species associations in an old-growth temperate forest in north-eastern China. *Journal of Ecology* 98: 674–686

Wang, X, T. Wiegand, N. G. Swenson, A. Wolf, R. Howe, and Z. Hao. 2015. Mechanisms underlying local functional and phylogenetic beta diversity in two temperate forests. Ecology 96: 1062–1073.

Wang, X., T. Wiegand, N.J.B. Kraft, N.G. Swenson, S.J. Davies, Z. Hao, R. Howe, Y. Lin, K. Ma, X. Mi, S-H Su, I-F Sun, and A Wolf. 2016. Stochastic dilution effects weaken deterministic effects of niche-based processes on the spatial distribution of large trees in species rich forests. *Ecology*

Webb CO, Ackerly DD, McPeek MA, Donoghue MJ. 2002. Phylogenies and community ecology. *Annu. Rev. Ecol. Syst.* 33:475–505

Wiegand T., and K. A. Moloney 2004. Rings, circles and null-models for point pattern analysis in ecology. *Oikos* 104: 209-229.

Wiegand, T, C.V.S. Gunatilleke, I.A.U.N. Gunatilleke, and T. Okuda. 2007a. Analyzing the spatial structure of a Sri Lankan tree species with multiple scales of clustering. Ecology 88: 3088–3102.

Wiegand, T, S. Gunatilleke, and N. Gunatilleke. 2007b. Species associations in a heterogeneous Sri Lankan Dipterocarp forest. *American Naturalist* 170 E77–E95.

Wiegand, T., C.V.S. Gunatilleke, I.A.U.N. Gunatilleke, and A. Huth. 2007c. How single species increase local diversity in tropical forests. *PNAS* 104:19029–19033.

Wiegand, T., A. Huth, S. Getzin, X. Wang, Z. Hao, C.V.S. Gunatilleke, and I.A.U.N. Gunatilleke. 2012. Testing the independent species arrangement assertion made by theories of stochastic geometry of biodiversity. Proceedings B 279: 3312-3320

Wiegand, T, F. He, and S.P. Hubbell. 2013. A systematic comparison of summary characteristics for quantifying point patterns in ecology. Ecography 36: 92-103.

Wiegand T., and K. A. Moloney 2014. *A handbook of spatial point pattern analysis in ecology*. Chapman and Hall/CRC press, Boca Raton, FL.

Wiegand, T., P. Grabarnik, and D. Stoyan. 2016. Envelope tests for spatial point patterns with and without simulation. Ecosphere 7(6):e01365

Wiegand, T., M. Uriarte, N.J.B. Kraft, G. Shen, X. Wang, and F. He. 2017. Spatially explicit metrics of species diversity, functional diversity, and phylogenetic diversity: insights into plant community assembly processes. *Annual Review of Ecology, Evolution, and Systematics* 48:329–351.

Wiegand, T., F. May, M. Kazmierczak and A. Huth. 2017b. What drives the spatial distribution and dynamics of local species richness in tropical forest. *Proceedings R. Soc. B* 284: 20171503

Wood, S.A. Statistical inference for noisy nonlinear ecological dynamic systems. 2010. Nature 466: 1102–1104.

Yang J, Swenson NG, Cao M, Chuyong GB, Ewango CEN, et al. 2013. A phylogenetic perspective on the individual species-area relationship in temperate and tropical tree communities. *PLoS One* 8(5): e63192