

# Introduction to Madagascar Software Project

An open-source software package for transparency in science!



**Yang Liu**

College of Geo-exploration Sciences and Technology

Jilin University

yangliu1979@jlu.edu.cn

Sep 6, 2020



1<sup>st</sup> AP  
GSC

6-10 Sept, 2020  
USTC, Hefei  
Anhui, China



# Outline

- **Science and Reproducibility**
- **Why Madagascar?**
- **Overview of Madagascar Software Project**
- **Madagascar Basic Usage**
- **Plans for Future Development**
- **Open-source Geophysics in China**



# Outline

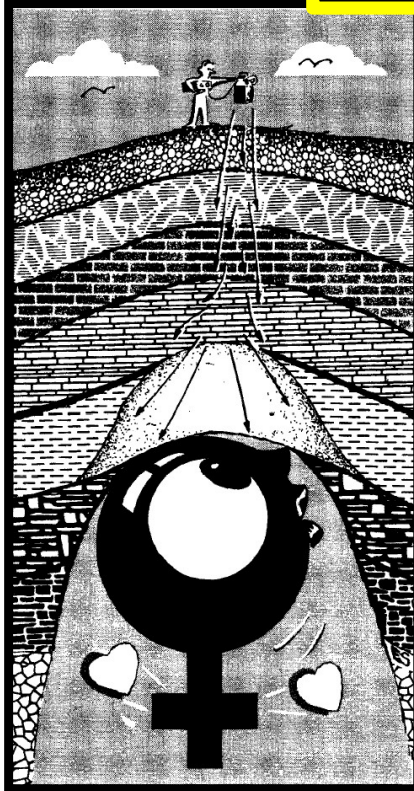
- **Science and Reproducibility**
- Why Madagascar?
- Overview of Madagascar Software Project
- Madagascar Basic Usage
- Plans for Future Development
- Open-source Geophysics in China

# Science or Black Magic ?

VOLUME I

JANUARY, 1936

NUMBER 1



## GEOPHYSICS

*A Journal of General and Applied Geophysics*

BLACK MAGIC IN GEOPHYSICAL PROSPECTING<sup>1</sup>

L. W. BLAU<sup>2</sup>

*EDITOR'S NOTE: The term "doodle-bug" is coming more and more to mean proposed methods of geophysical prospecting that are neither based upon scientific fact nor upon known or proven properties of oil, minerals and geologic formations. The geophysicist is often consulted concerning the reliability of such a proposed method, and his task then is to explain scientifically just why the proposed method fails and is unsuitable for the intended purpose.*

Slice courtesy of Sergey Fomel (ICERM, Providence, 2012)

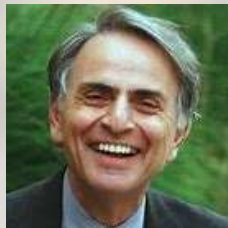
# What is Science ?

The word science comes from the Latin "scientia", meaning knowledge.

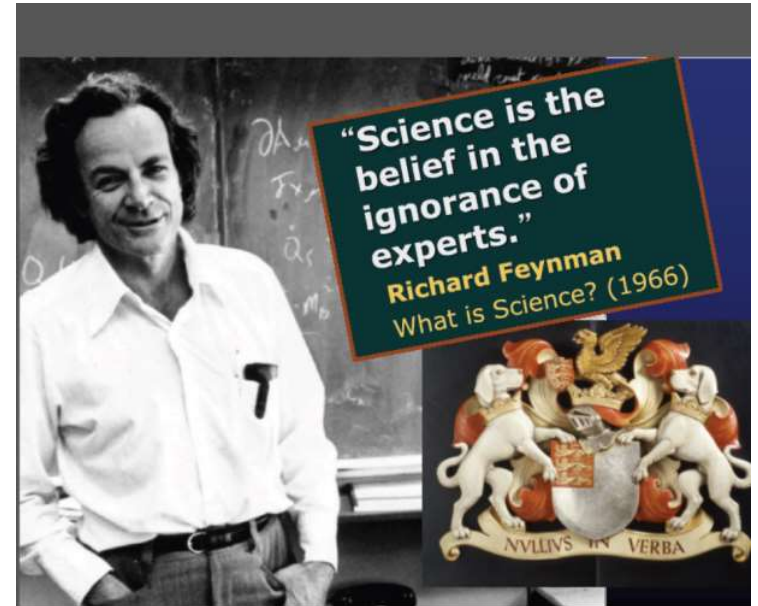
**SCIENCE IS A  
WAY OF  
THINKING  
MUCH MORE  
THAN IT IS A  
BODY OF  
KNOWLEDGE.**

Carl Sagan

PICTUREQUOTES.com



**Science** is the systematic enterprise of gathering knowledge about the universe and organizing and condensing that knowledge into testable laws and theories. The success and credibility of science are anchored in the willingness of scientists to expose their ideas and results to **independent testing and replication** by other scientists. This requires the **complete and open exchange of data, procedures and materials.**



Courtesy of Sergey Fomel (School, Shanghai, 2017)

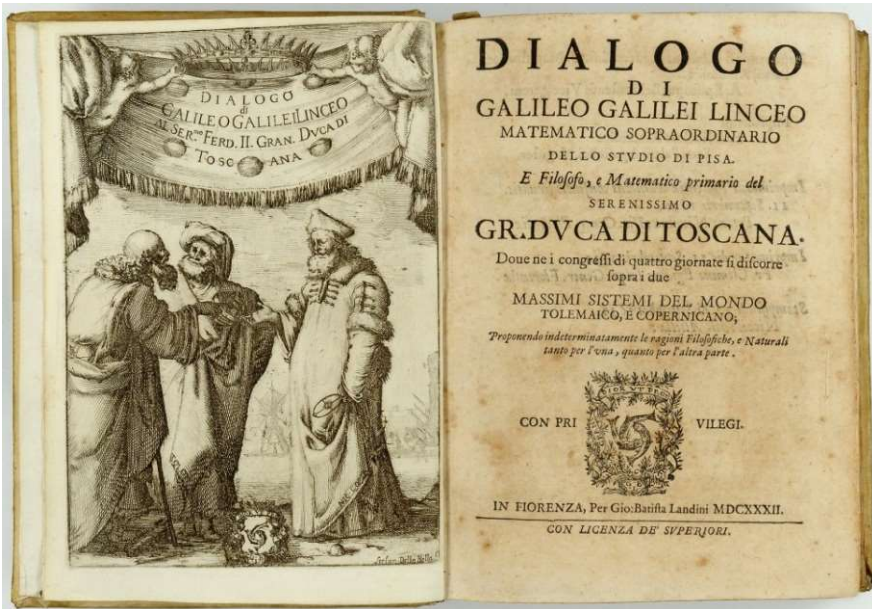
# Why Reproducible?

...if you are publishing research articles that use computer programs, if you want to claim that you are engaging in **science**, the programs are in your possession and you will not **release** them then I would not regard you as a scientist; I would also regard any papers based on the software as null and void.

--From 《The Guardian》

the guardian

**Science needs vocal skeptics and openness!!**



ROYAL SOCIETY  
OPEN SCIENCE

[rsos.royalsocietypublishing.org](https://rsos.royalsocietypublishing.org)

Editorial



Cite this article: Sanders *et al.* 2017  
Transparency and openness in science *R. Soc. open sci.* **4**: 160979.  
[http://dx.doi.org/10.1098/rsos.160979](https://dx.doi.org/10.1098/rsos.160979)

Transparency and openness  
in science

Jeremy Sanders<sup>1</sup>, Jon Blundy<sup>2</sup>, Anne Donaldson<sup>3</sup>, Steve Brown<sup>4</sup>, Rob Ivison<sup>5,6</sup>, Miles Padgett<sup>7</sup>, Kevin Padian<sup>8</sup>, Katrin Rittinger<sup>9</sup>, Kerry Rowe<sup>10</sup>, Anthony Stace<sup>11</sup>, Essi Viding<sup>12</sup>, Chris Chambers<sup>13</sup> and Mark Chaplain<sup>14</sup>

<sup>1</sup>Department of Chemistry, University of Cambridge, Cambridge, CB2 1EW, UK

<sup>2</sup>School of Earth Sciences, University of Bristol, Bristol, UK

<sup>3</sup>Institute of Medical Sciences, University of Aberdeen, Aberdeen AB25 2ZD, UK

**Citation standards.** Level 2: article provides appropriate citation for data, code and materials used consistent with journal's author guidelines.

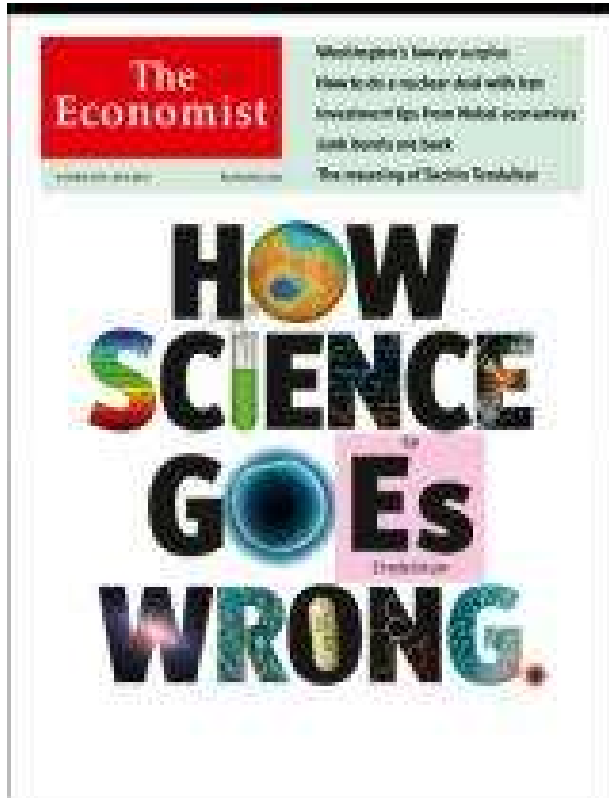
**Data transparency.** Level 2: data must be posted to a trusted repository. Exceptions must be identified at article submission.

**Analytic methods (code) transparency.** Level 2: novel code must be posted to a trusted repository. Exceptions must be identified at article submission.

**Digital research materials transparency.** Level 2: article states whether digital materials are available and, if so, where to access them.

# Why Reproducible?

**Science needs verification!!**



“A SIMPLE idea underpins science: **trust, but verify.** Result should always be subject to challenge from experiment. That simple but powerful idea has generated a vast body of knowledge. Since its birth in the 17th century, modern science has changed the world beyond recognition, and overwhelmingly for the better. But success can breed complacency. Modern scientists are doing too much trusting and not enough verifying to the detriment of the whole of science, and of humanity.”

<http://ahay.org/blog/2013/10/26/trust-but-verify/>

# Reproducible Research and Open-Source Software

## What is Reproducible Research?

- ▶ Attaching software code and data to publications
- ▶ Communicating computational results to a skeptic

*An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures. Jon Buckheit and David Donoho, WaveLab*

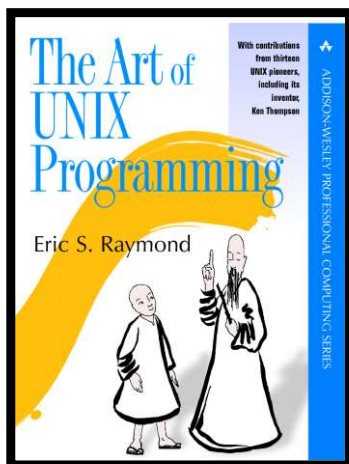
## Reproducible Research Discussions

▶ <http://www.reproducibleresearch.net>



Geophysicists call for papers  
Reproducible research:  
Geophysicists papers of the future

## From Science to Open-Source Software



“**Abandoning the habit of secrecy** in favor of process transparency and peer review was the crucial step by which alchemy became chemistry.

In the same way, it is beginning to appear that **open-source** development may signal the long-awaited maturation of software development as a discipline.”

Eric S. Raymond, TAUP, 2004

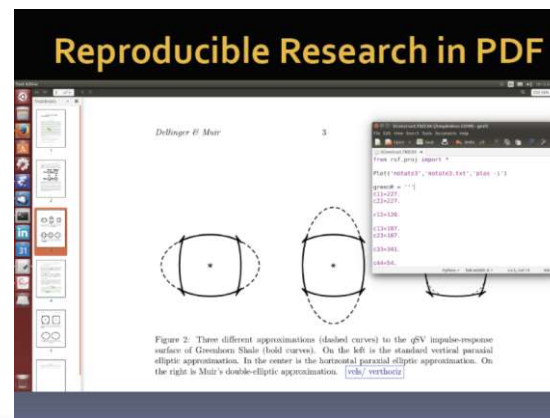
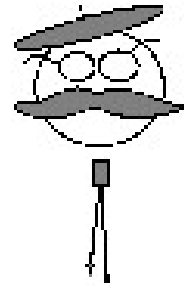
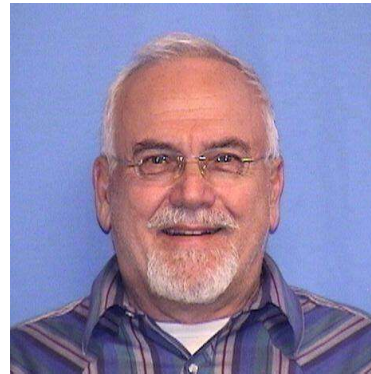
Courtesy of Sergey Fomel (SIAM Geosciences, Long Beach, 2011)



# Reproducible Principle

“It is a big chore for one researcher to reproduce the analysis and computational results of another [...] I discovered that this problem has a simple technological solution: illustrations (figures) in a technical document are made by **programs and command scripts that along with required data should be linked to the document itself** [...] This is hardly any extra work for the author, but it makes the document much more valuable to readers who possess the document in electronic form because they are able to track down the computations that lead to the illustrations.”

**(Claerbout, 1991)**



## Claerbout's Principle

“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is **the complete software development environment and the complete set of instructions which generated the figures.**”

**(Buckheit and Donoho, 1995)**

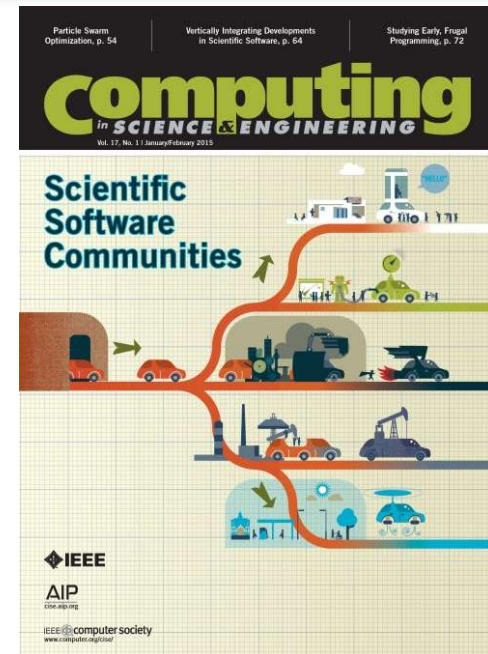
# Reproducible Laws

## Reproducible Research as a Community Effort: Lessons from the Madagascar Project

Sergey Fomel | University of Texas at Austin

Instead of computational reproducibility being the responsibility of an individual author, it should become the responsibility of open source scientific-software communities. A dedicated community effort can keep a body of computational research alive by actively maintaining its reproducibility. The Madagascar open source software project offers an example of such a community.

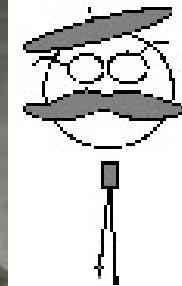
- **Reproducibility is not the goal.**
  - **The principal beneficiary is the author.**
  - **Each computation is a test.**
  - **Reproducibility requires maintenance, maintenance requires an open community.**
- (Fomel, 2015)**



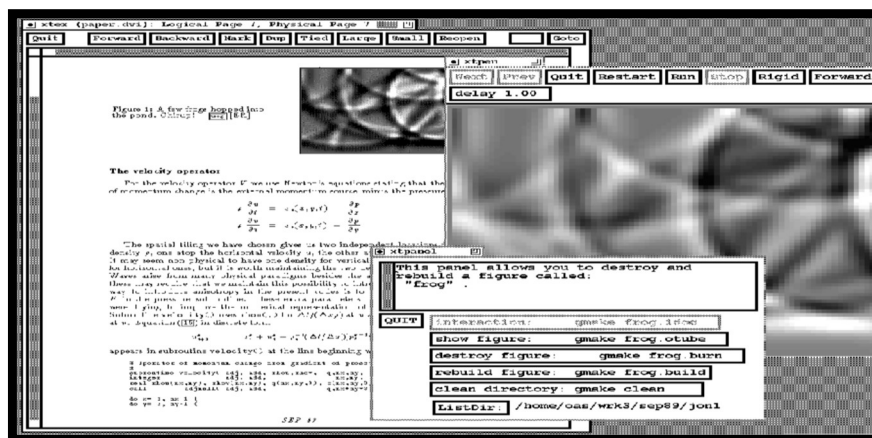
# Reproducible Research at SEP

## Jon Claerbout's Story

- **Stanford Exploration Project (SEP)**
  - Founded in 1973
  - 2 Ph.D. students per year
- **Reproducible research (SEPlib)**
  - Jon Claerbout constructed first version of SEPlib
  - From CD-ROMs to WWW
  - From cake to GNU make
  - 2001 *CiSE* paper



- **1987: Sunview experience**
    - Interactive programs are slavery
  - **1992: LaTeX + cake**
    - Rebuilding books by a single command
  - **1990s: Ph.D. students**
    - cake to make, CD-Rom to WWW
  - **2001: reproducible research paper in *CiSE***
    - The principal beneficiary is the author
- J. Claerbout, Electronic Documents Give Reproducible Research a New Meaning, in Proc. 62nd Ann. Int. Meeting of the Soc. of Exploration Geophysics, 1992, pp. 601–604



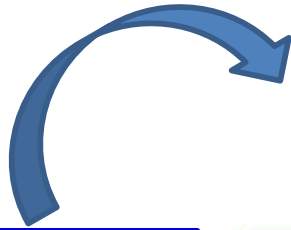


# Outline

- Science and Reproducibility
- **Why Madagascar?**
- Overview of Madagascar Software Project
- Madagascar Basic Usage
- Plans for Future Development
- Open-source Geophysics in China

# Why Madagascar ?

## Standalone-style suites – relative differences



Madagascar
SEPlib
SU
FreeUSP, FreeDDS
Pseis
CPSeis
SPARC
BotoSeis
GEBR
SeaSeis
JavaSEIS

Seismic Unix	Madagascar	SEPlib
<ul style="list-style-type: none"> <li>• Many utilities for preprocessing, especially <b>irregularly-sampled</b> data handling</li> <li>• <b>Few bugs</b> (large user base)</li> <li>• <b>Stable</b></li> <li>• Data format more suited to preprocessing than imaging</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Few bugs</b> (test-driven!), <b>maintainable</b>, <b>portable</b>, active community</li> <li>• Still <b>evolving</b> towards better <b>functionality</b> and even higher <b>robustness</b>, <b>portability</b> and <b>maintainability</b> (this is good, but <b>less stable</b>)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Stable</b>, but <b>buggy</b> (small user base)</li> <li>• <b>Irregularly-sampled</b> data handling in RSF-compatible format</li> <li>• <b>Backwards compatibility</b> with legacy imaging workflows</li> </ul>

Courtesy of Ioan Vlad and Charles Kovacs (OSTRCG, Houston, 2011)

# Why Madagascar ?

Madagascar is not the only open-source geophysical package, but it is the only one to provide all the components of a shared research environment:



- Standalone **programs** for out-of-core data analysis;
- Standalone **programs** for geophysical data processing and imaging;
- A **development kit** for C, C++, Java, Fortran-77, Fortran-90, Python, Matlab, Octave, and Julia;
- A framework for **reproducible** numerical **experiments**, based on [SCons](#);
- A framework for **scientific publications**, based on [SCons](#) and [LaTeX](#);
- A collection of **reproducible** scientific **articles** also used as usage examples and regression tests for the standalone programs;
- A collection of **datasets** used as input to reproducible numerical experiments.



# Outline

- Science and Reproducibility
- Why Madagascar?
- **Overview of Madagascar Software Project**
- Madagascar Basic Usage
- Plans for Future Development
- Open-source Geophysics in China

# Madagascar Software Project

The Leading Edge® JOURNAL HOME CURRENT ISSUE ISSUES PODCASTS DIGITAL EDITION SUBSCRIBE SUBMISSIONS ABOUT ▾

PREVIOUS NEXT

FIGURES REFERENCES RELATED DETAILS

You have access | The Leading Edge | Volume 34, Issue 11

## OPEN-SOURCE SOFTWARE

# The genesis of Madagascar

John Holden<sup>1</sup>

<https://doi.org/10.1190/tle34111386.1>

Sections | Abstract | PDF/EPUB | Tools | Share

Volume 34  
Issue 11  
Nov 2015  
Pages: 1298-1424  
ISSN (print): 1070-485X  
ISSN (online): 1938-3789

### PUBLICATION DATA



Locations of authors contributing reproducible paper  
(2006-2015)

- Project manager: Prof. Sergey Fomel
- Start around 2003
- Publicly available since June 12, 2006
- Current stable version: 3.0
- 84 developers and more by now
- 25 workshops and schools in 10 countries
- Main Page: <http://www.ahay.org>



# Madagascar History



## 1. About name:

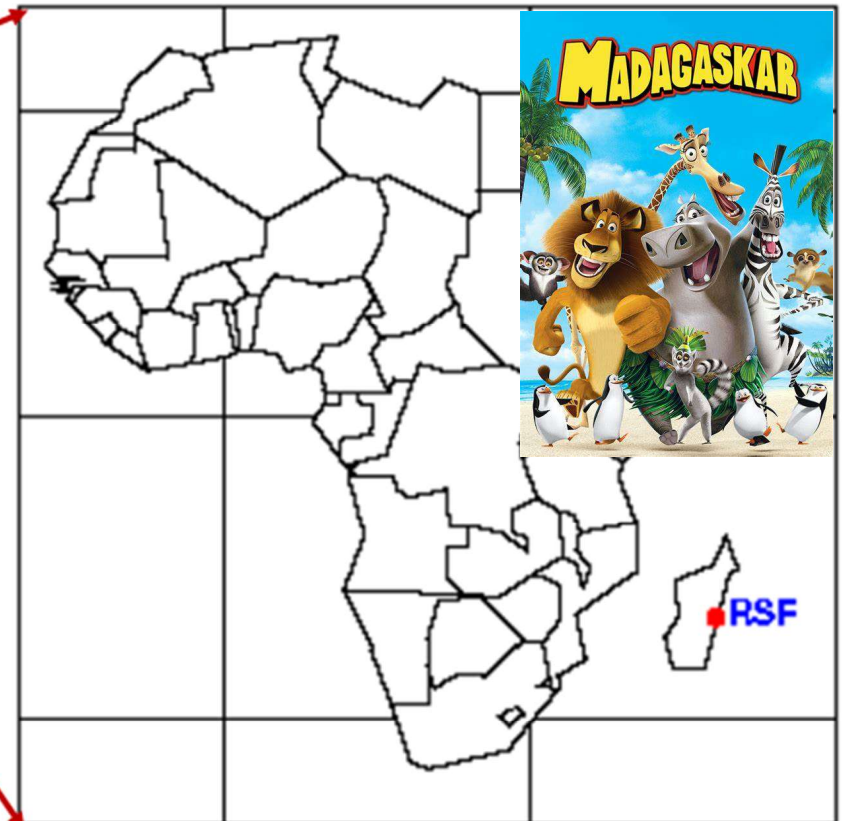
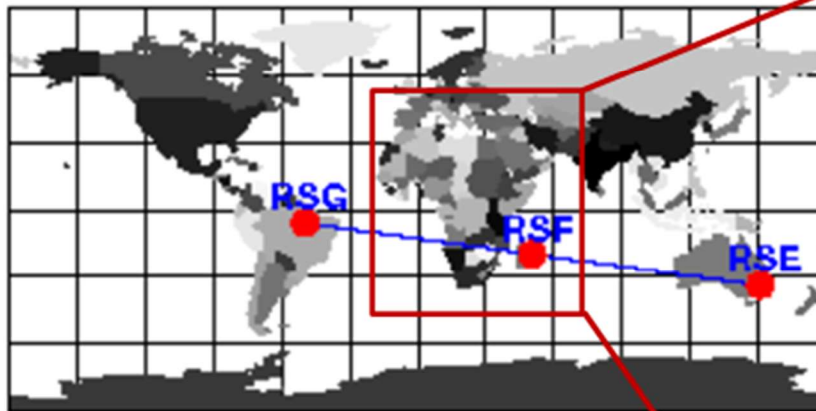
**RSF (The Regularly Sampled Format):** A meaningless abbreviation that nobody can remember.

**Madagascar:** A symbol of isolation, which is a reminder of what RSF does not want to be.

**Ahay:** the Malagasy name for aye-aye, a strange nocturnal primate that lives in Madagascar.

RSE: Sydney Rose Bay in Australia

RSG: Serra Pelada in Brazil



<http://ahay.org/blog/2006/04/19/madagascar/>



Prof. Sergey Fomel created a virtual international airport codes for Madagascar software.

# Madagascar History

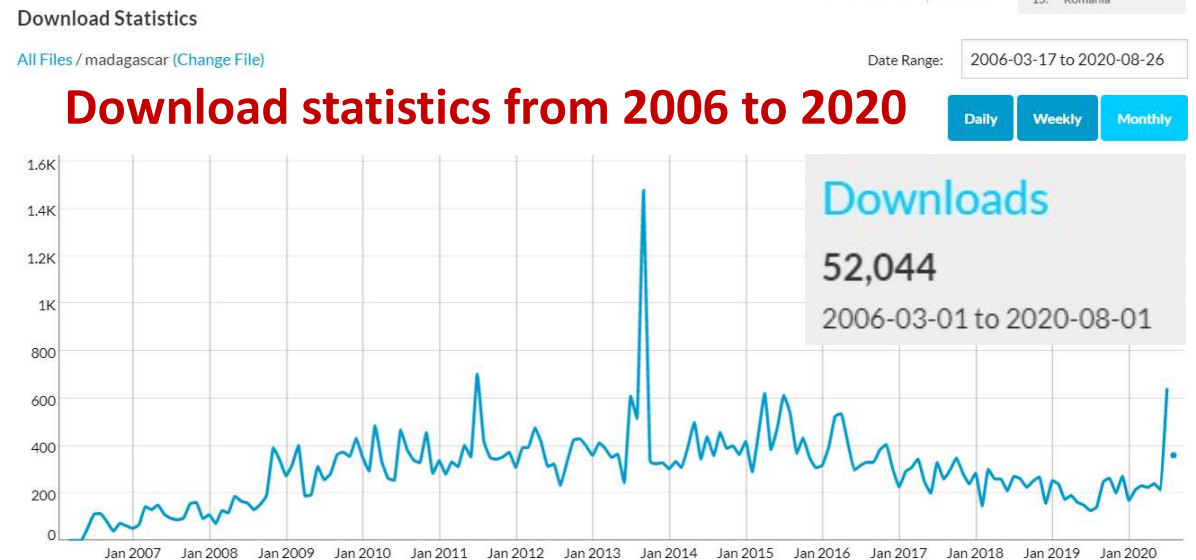
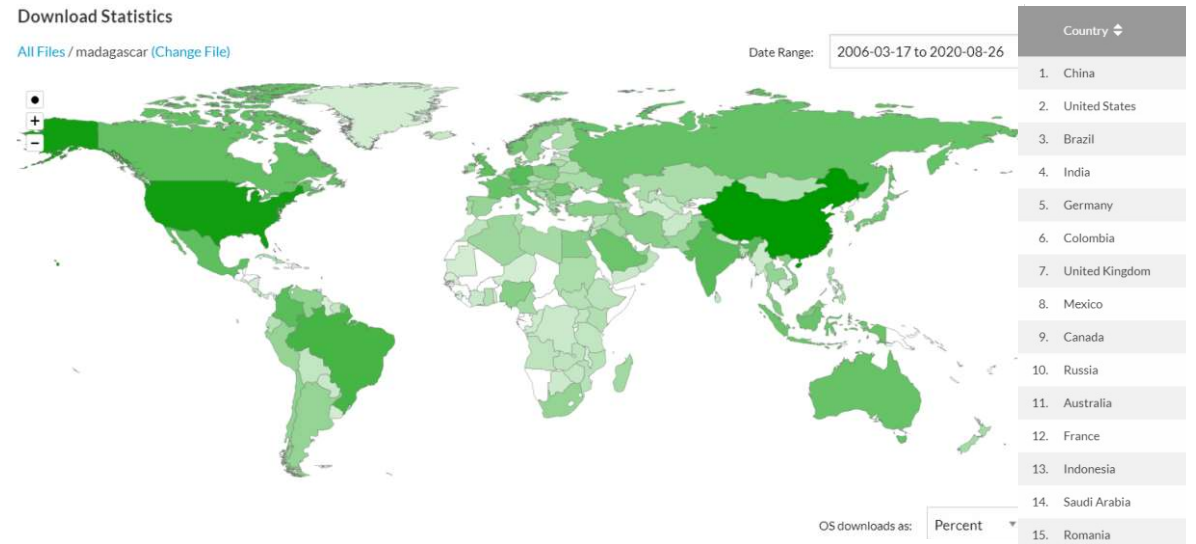
## 2. Release notes for stable version:

[http://ahay.org/wiki/Release\\_Notes#Release\\_2.0\\_.282017-07-08.29](http://ahay.org/wiki/Release_Notes#Release_2.0_.282017-07-08.29)

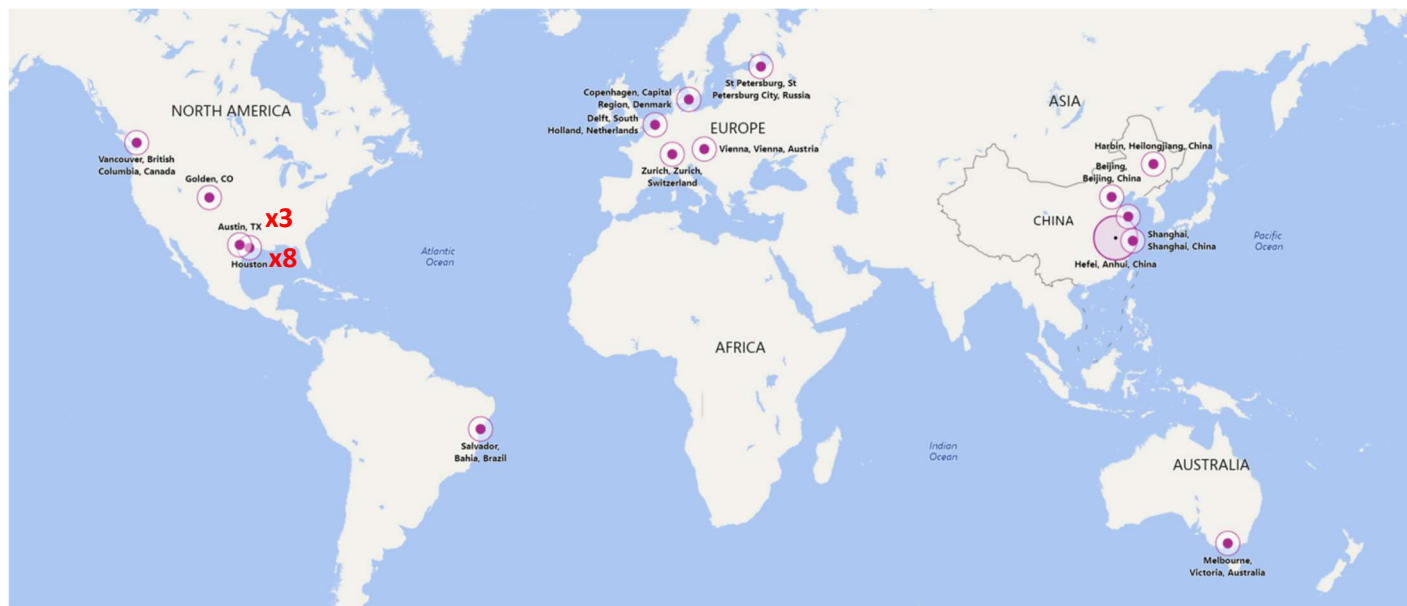
- ✓ **Release 1.0 (2010-07-23)**
- ✓ Release 1.1 (2011-01-18)
- ✓ Release 1.2 (2011-07-14)
- ✓ Release 1.3 (2012-04-22)
- ✓ Release 1.4 (2012-04-22)
- ✓ Release 1.5 (2013-07-24)
- ✓ Release 1.6 (2014-05-09)
- ✓ Release 1.7 (2014-04-13)
  
- ✓ **Release 2.0 (2017-07-08)**
- ✓ **Release 3.0 (2019-09-09)**

## 3. Major administrative change:

- ✓ **Repository (2015-7-28) :**  
From SourceForge to GitHub
- ✓ **Blog (2015-8-29) :**  
From Serendipity to WordPress



# Madagascar Activities



Locations of Madagascar workshops and schools 2006–2020

## Conference Presentations

- Lausanne 2016 (PASC)
- San Francisco 2015 (AGU)
- Stanford 2015 (SIAM Geosciences)
- New York 2013 (NYU Poly)
- Providence 2012 (ICERM)
- Austin 2012 (SciPy)
- Cary 2011 (Interface)
- Long Beach 2011 (SIAM Geosciences)
- Reno 2011 (SIAM CS&E)
- Austin 2010 (SciPy)
- Salt Lake City 2010 (NSF Archive Workshop)
- Düsseldorf 2008 (Berlin 6)
- Austin 2008 (Texas Python Unconference)
- Austin 2008 (Scientific Software Days)
- Rio de Janeiro 2007 (SBGf)
- Vancouver 2007 (AIP)
- Honolulu 2007 (ICASSP)
- Austin 2007 (Scientific Software Day)
- New Orleans 2006 (SEG)
- Vienna 2006 (EAGE)

## Madagascar Workshops and Schools

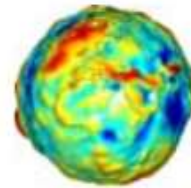
- 2006--Vancouver, Canada
- 2007--Austin, USA
- 2008--Golden, USA
- 2009--Delft, Netherlands; Salvador, Brazil
- 2010--Houston, USA
- 2011--Beijing, China; Houston, USA
- 2012--Austin, USA; Copenhagen, Denmark
- 2013--Melbourne, Australia; Austin, USA
- 2014--St. Petersburg, Russia; Houston, USA
- 2015--Harbin, China ; Houston, USA; Qingdao, China
- 2016--Zürich, Switzerland; Vienna, Austria; Houston, USA
- 2017--Houston, USA; Shanghai, China; Houston, USA
- 2018--Houston, USA
- 2020--Hefei, China (current)

# Madagascar Architecture



## SOFTWARE PAPERS

### Madagascar: open-source software project for multidimensional data analysis and reproducible computational experiments



Sergey Fomel,<sup>1</sup> Paul Sava,<sup>2</sup> Ioan Vlad,<sup>3</sup> Yang Liu,<sup>4</sup> Vladimir Bashkardin,<sup>5</sup>

<sup>1</sup> Jackson School of Geosciences, The University of Texas at Austin, Austin, Texas, USA

<sup>2</sup> Center for Wave Phenomena, Colorado School of Mines, Golden, Colorado, USA

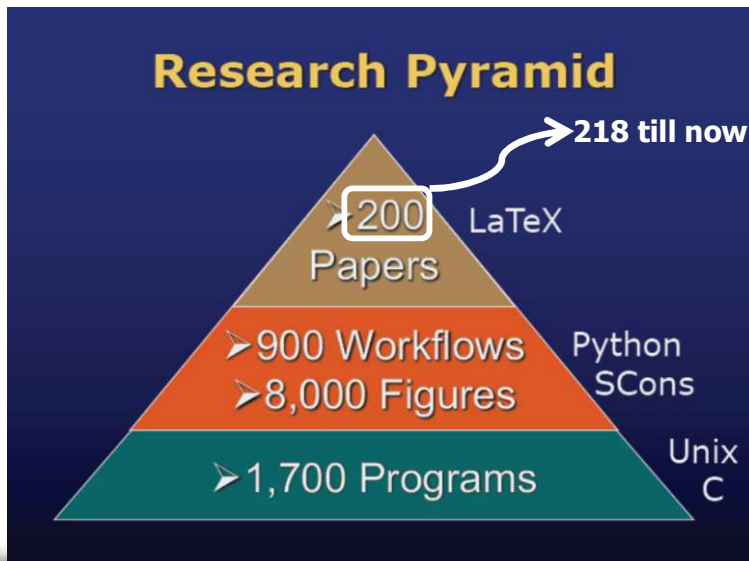
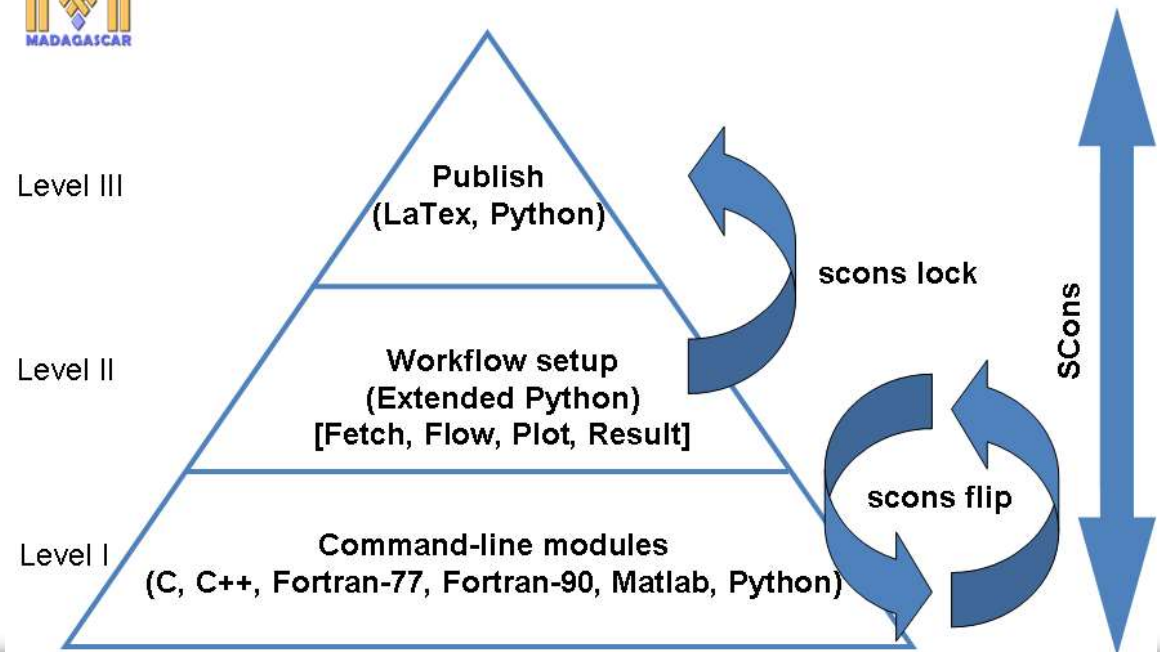
<sup>3</sup> TGS, Houston, Texas, USA

<sup>4</sup> College of Geo-exploration Science and Technology, Jilin University, Changchun, Jilin, China

<sup>5</sup> BP, Houston, Texas, USA



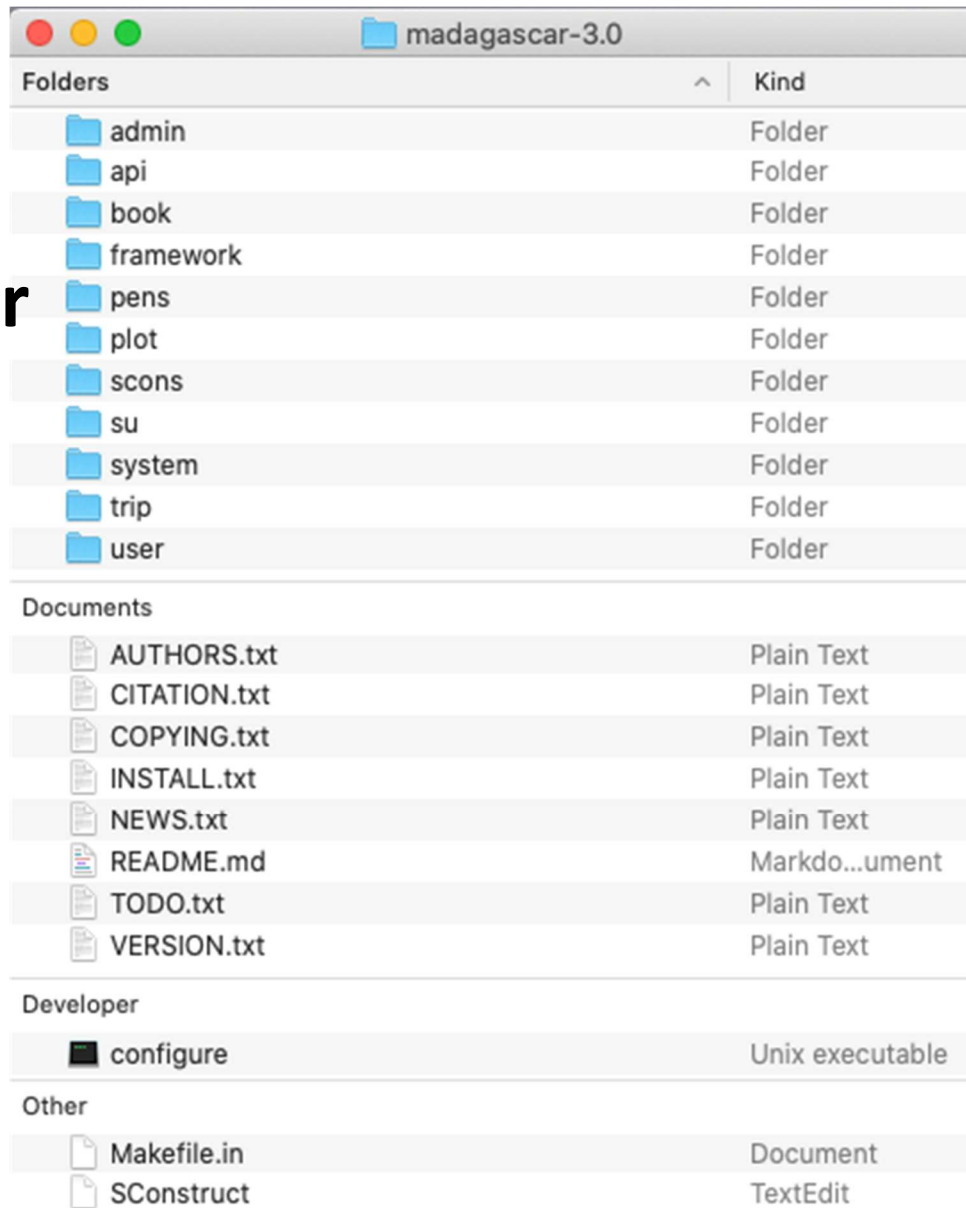
### Madagascar software architecture



Courtesy of Sergey Fomel (School, Houston, 2017)

# Software Structure

## Madagascar Source Directory Structure



Folders	Kind
admin	Folder
api	Folder
book	Folder
framework	Folder
pens	Folder
plot	Folder
scons	Folder
su	Folder
system	Folder
trip	Folder
user	Folder

Documents	Kind
AUTHORS.txt	Plain Text
CITATION.txt	Plain Text
COPYING.txt	Plain Text
INSTALL.txt	Plain Text
NEWS.txt	Plain Text
README.md	Markdo...ument
TODO.txt	Plain Text
VERSION.txt	Plain Text

Developer	Kind
configure	Unix executable

Other	Kind
Makefile.in	Document
SConstruct	TextEdit

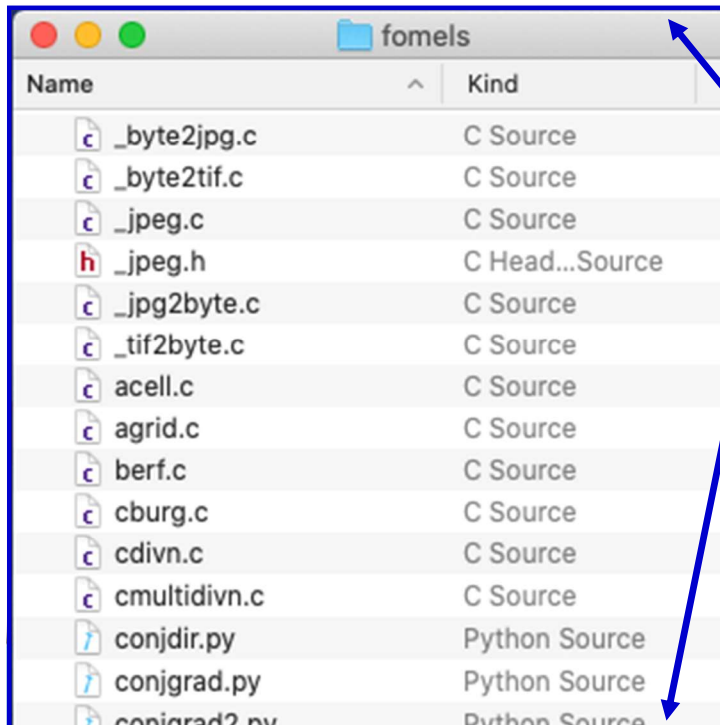
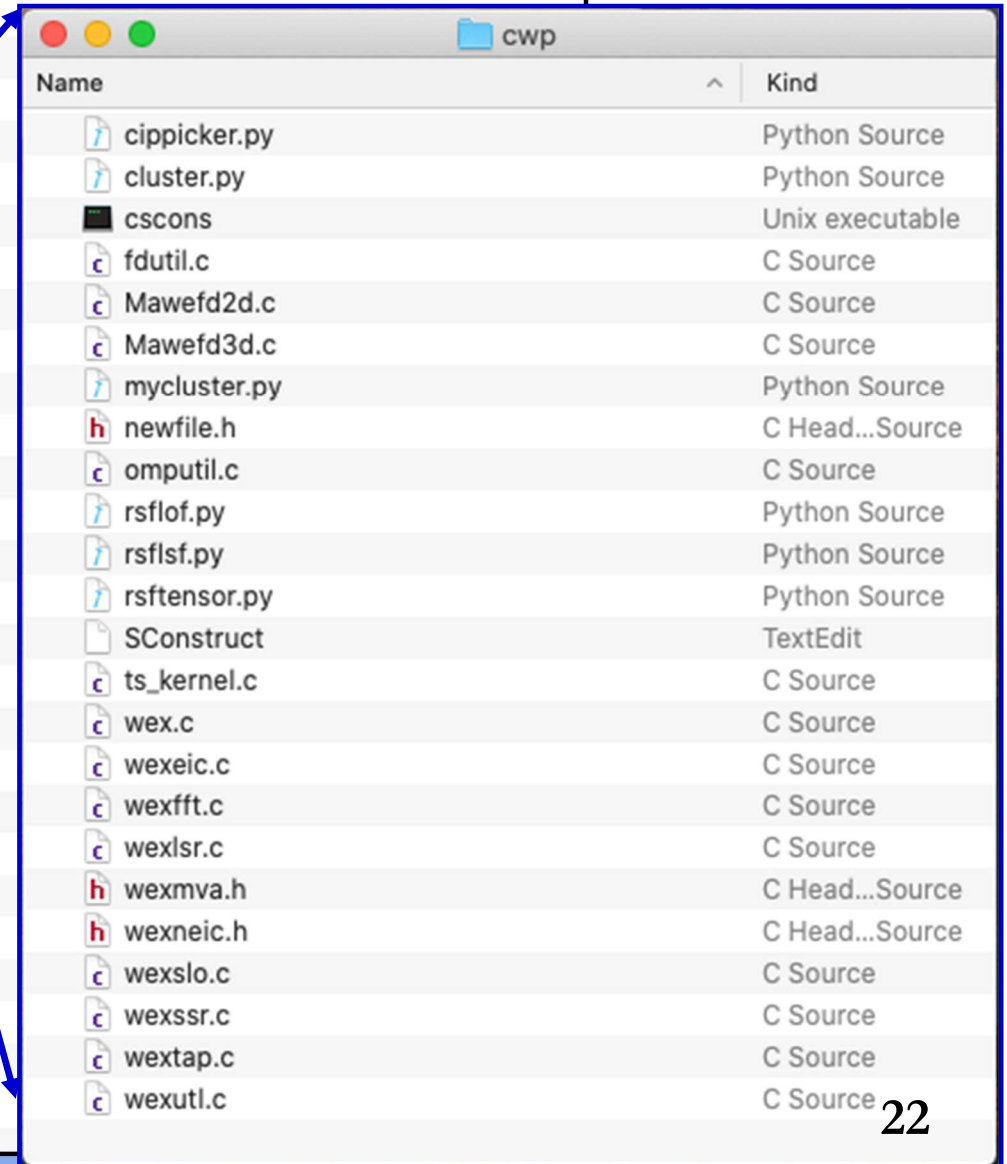
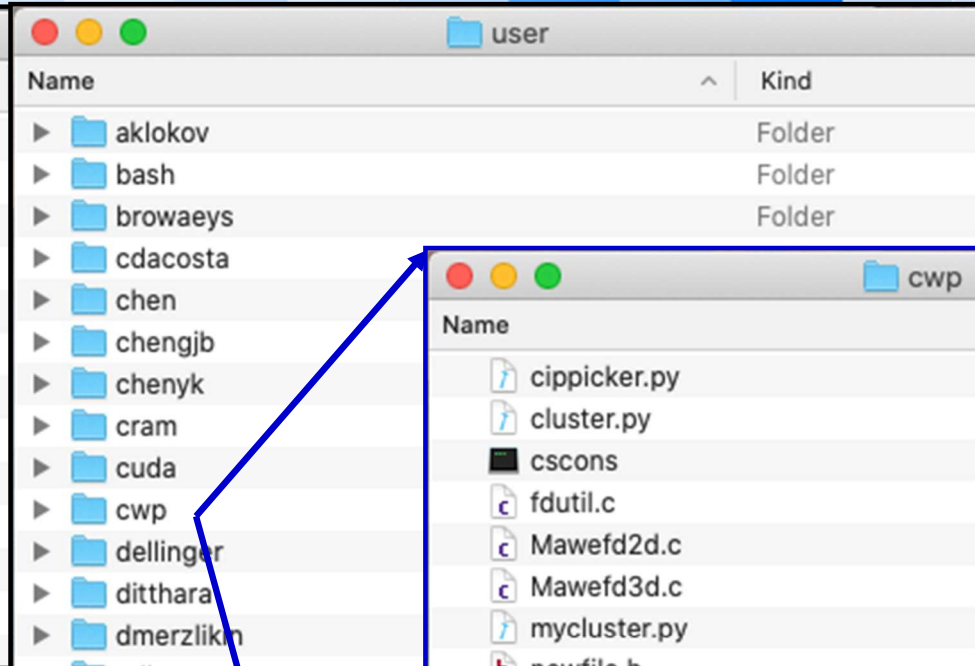
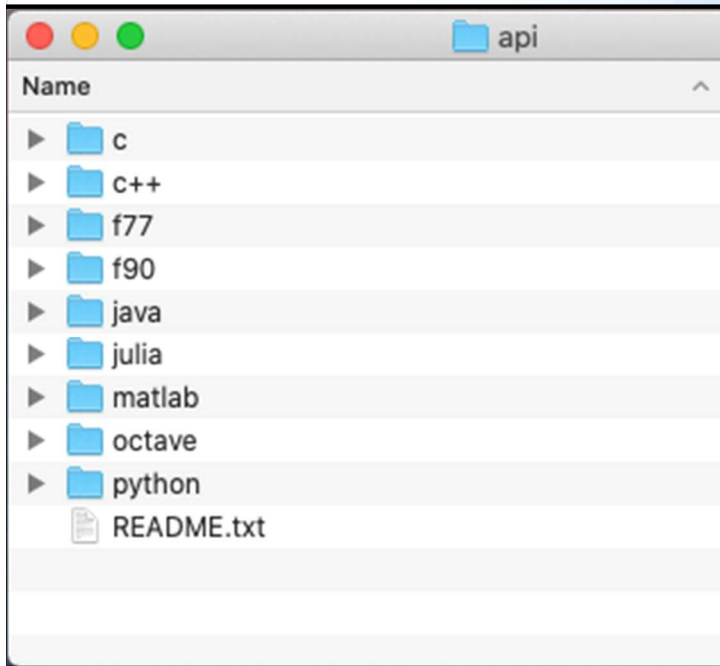
## Description

**Administrative and maintenance tools**  
**Programming Interfaces**  
**Reproducible documents**  
**RSF Python package**  
**Sources for Vplot manipulation**  
**Sources for Vplot generation**  
**SCons sources**  
**SU codes in Madagascar**  
**Source codes for core programs**  
**Codes from The Rice Inversion Project**  
**Source codes from Developers**

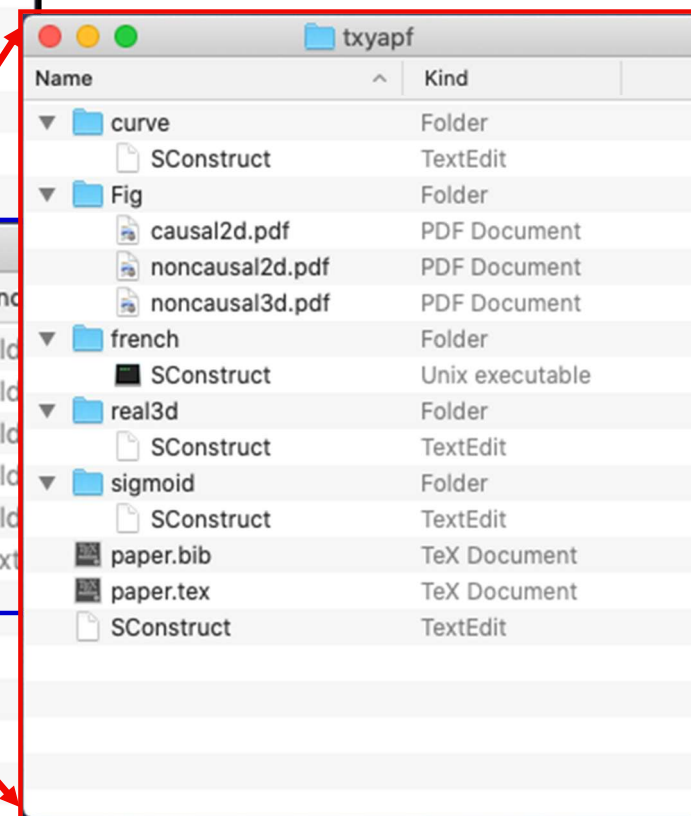
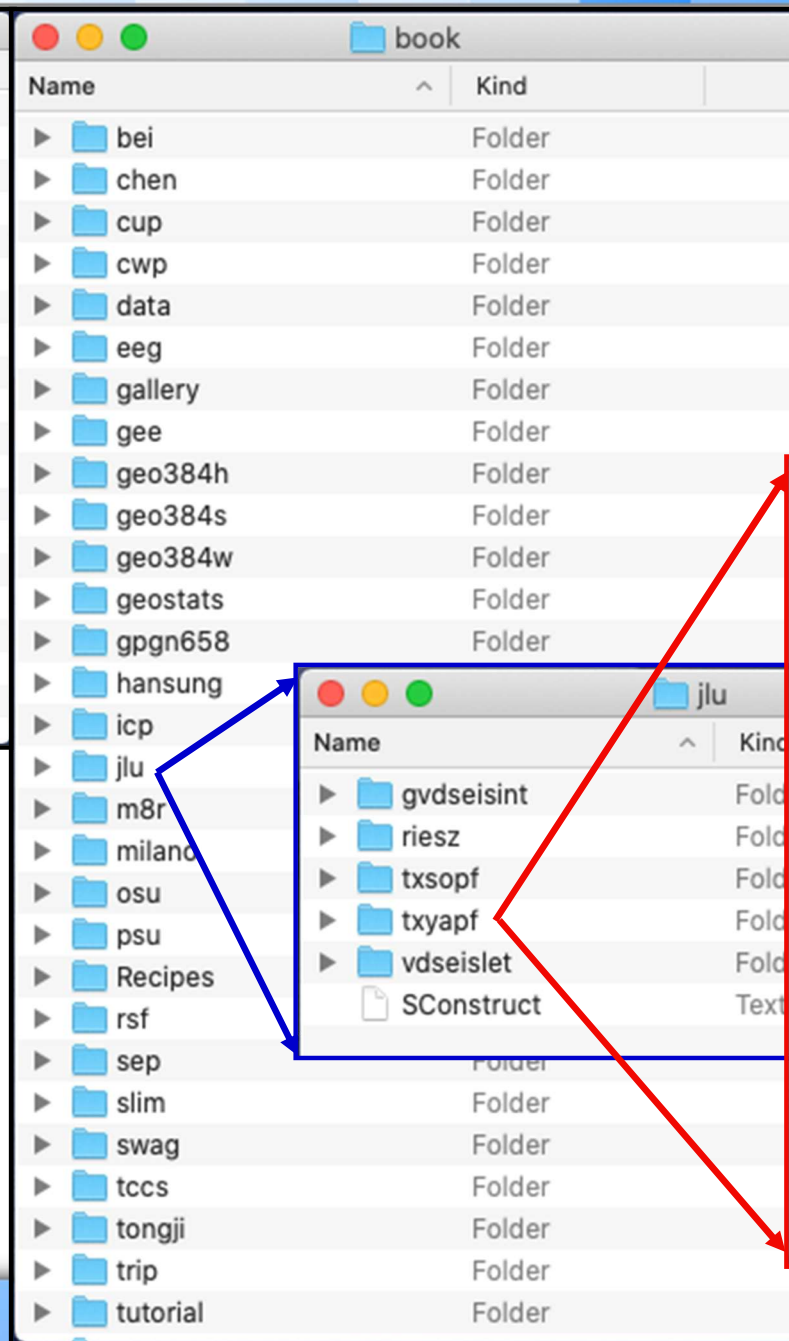
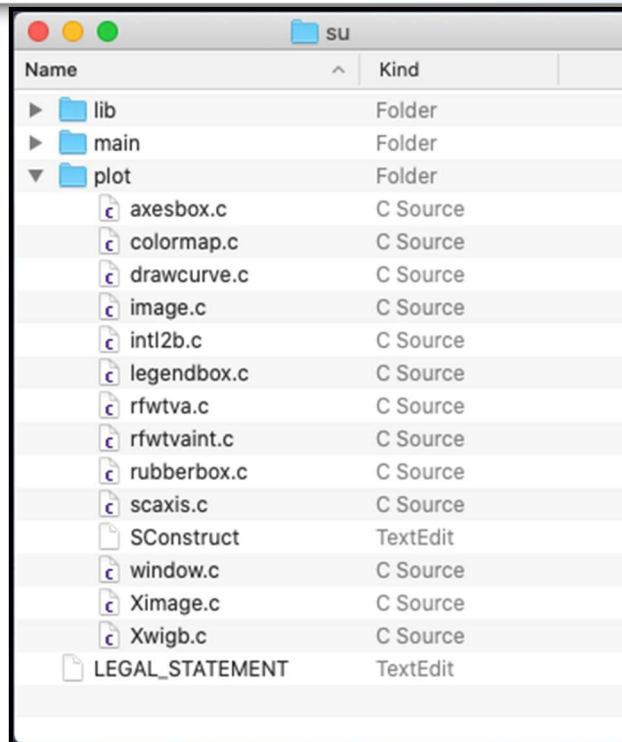
**Top software construction**

# Look inside I

## Standalone programs



# Look inside II

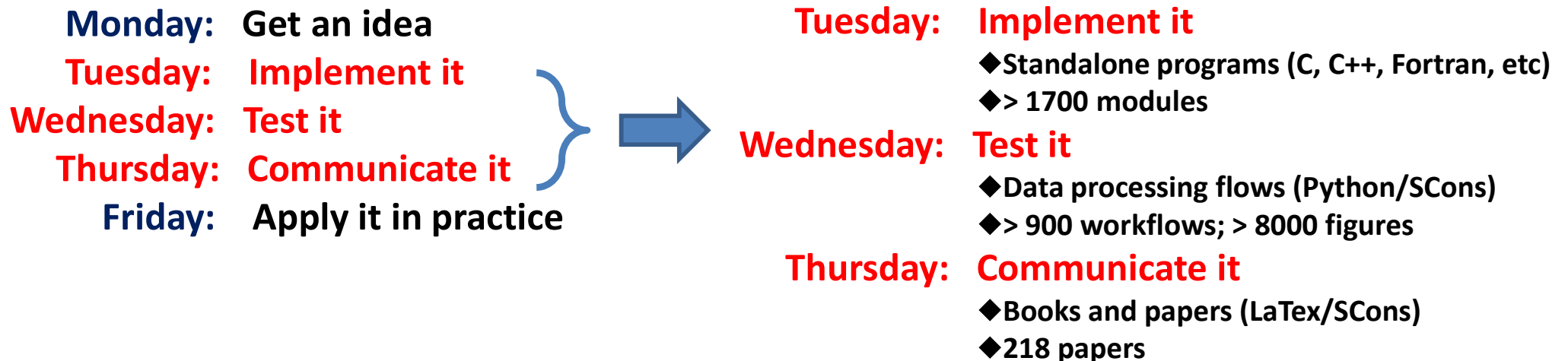


# Madagascar components:

[http://www.ahay.org/wiki/Package\\_overview#Madagascar\\_components](http://www.ahay.org/wiki/Package_overview#Madagascar_components)

- Standalone programs
- Data format
- Reproducible documents
- Vplot graphics
- Reproducibility and Project Management

## One Week Technology Transfer:



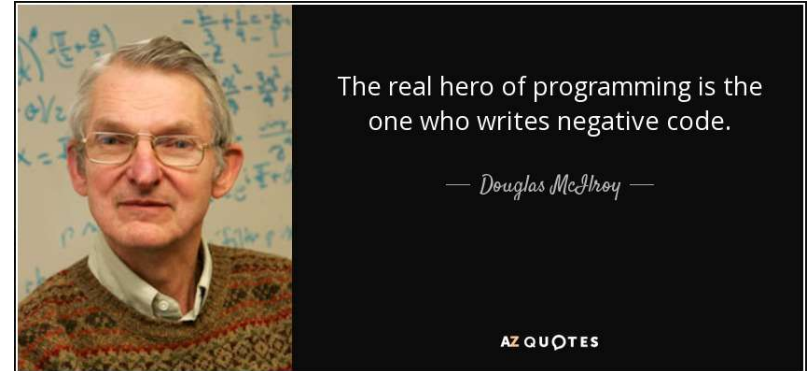


# Madagascar Design Principles

## 1. **Modularity.**

“Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that it a universal interface.”

-- Doug McIlroy



## 2. **KISS.**

“Keep It Simple, Stupid!”

## 3. **Test-driven development.**

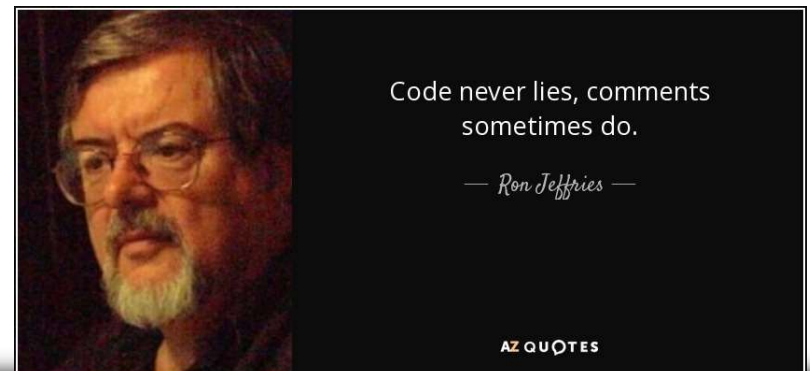
A Madagascar modules is not included in the official distribution until there is an example of its usage in reproducible documents.

## 4. **YAGNI.**

“You Ain’t Gonna Need It!”

“Always implement things when you actually need them, never when you just foresee that you need them.”

-- Ron Jeffries





# Outline

- Science and Reproducibility
- Why Madagascar?
- Overview of Madagascar Software Project
- **Madagascar Basic Usage**
- Plans for Future Development
- Open-source Geophysics in China

# Madagascar Basic Usage

## 1. Formats:

### ➤ Data format: RSF

- ◆ Simple universal data file format Borrowed from SEPLib
- ◆ Data separated from text headers

#### Header (\*.rsf):

Text file (description of data)  
Description of regularly-sampled format  
Small, can be archived

#### Binary (\*.rsf@):

Binary file (actual data)  
N-dimensional (up to 9) hypercube  
Large, can be stored on a different file system  
Path to binary set with environment variable DATAPATH

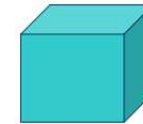
- ◆ Multidimensional arrays as files
- ◆ Multiple files for irregular data
- ◆ Compatibility with other file formats (SEPLib, SEG Y, SU, ASCII, CSV, jpg):

[http://www.ahay.org/wiki/Guide\\_to\\_RSf\\_file\\_format#Compatibility\\_with\\_other\\_file\\_formats](http://www.ahay.org/wiki/Guide_to_RSf_file_format#Compatibility_with_other_file_formats)

### ➤ Figure format: Vplot (\*.vpl)

- ◆ Circa 1979, from the idea of a “virtual device”.
- ◆ Joe Dellinger wrote a whole new and excellent routine for reading Vplot.

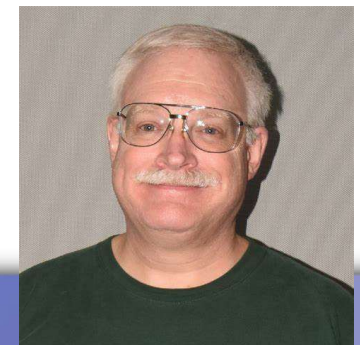
<http://reproducibility.org/wikilocal/docs/Vplot2010.ppt>



```
n1=1000 in="/path/data.rs@"  
n2=500 n3=100 d1=0.001 d2=0.1 o2=1
```

Data

Data Type	Data Form
<ul style="list-style-type: none"><li>• uchar,</li><li>• char</li><li>• int</li><li>• float</li><li>• complex</li><li>• short</li><li>• double</li></ul>	<ul style="list-style-type: none"><li>• native</li><li>• XDR</li><li>• ASCII</li></ul>



# Madagascar Basic Usage

## 2. Command-line usage:

[http://www.ahay.org/wiki/Tutorial#Command\\_line\\_interface](http://www.ahay.org/wiki/Tutorial#Command_line_interface)

### ➤ Program name uses prefix “sf”

- ◆ List of all programs by using “sfdoc -k .”
- ◆ Look for specific programs by using “sfdoc -k <keyword>”

Demo:

**sfdoc -k median**

```
smedianbalance: Do median balancing.
sfwmf: 1D weighted median filtering.
sftvmf: 1D Time-varying median filtering.
sfmf: 1D median filtering.
sflrmf: Local radial median filtering.
sfdespike: Remove spikes in by sliding 1-D medians.
sfdespike2: Remove spikes in by sliding 2-D medians.
sfdespike3: Remove spikes in by sliding 3-D medians.
sfmlwm: 2D Multistage weighted median filtering.
sfdomf: 2D dip-oriented median/mean filter (DOMF).
sfmedian: Compute median on the first axis.
sfsvmf: Space varying median filtering.
sfmlm: 2D Multistage median filtering.
sftsmf: Two-step space varying median filtering.
sfnlm1: 1D non-local median filtering.
sfnlm2: 2D non-local median filtering.
sfdownmf: 2D dip-oriented weighted median filter (DOWMF).
```

# Madagascar Basic Usage

## ➤ Self-documentation

- ◆ Run program without arguments
- ◆ Find parameters, examples, and source code and run demo.

**Example:**

**Demo:**

**sfspike**

**sfspike n1=100 o1=0 d1=0.01 n2=50 o2=1000 d2=10 > file1.rsf**

```
NAME
    sfspike
DESCRIPTION
    Generate simple data: spikes, boxes, planes, constants.
SYNOPSIS
    sfspike < in.rsf > spike.rsf mag= nsp=1 k#[0,...] l#[k1,k2,...] p#
    =[0,...] n#= o#[0,0,...] d#[0.004,0.1,0.1,...] label#[Time,Distance,Dista
    nce,...] unit#[s,km,km,...] title=
COMMENTS
    Spike positioning is given in samples and starts with 1.
PARAMETERS
    float  d#[0.004,0.1,0.1,...]  sampling on #-th axis
    ints   k#[0,...]              spike starting position [nsp]
    ints   l#[k1,k2,...]         spike ending position [nsp]
    string label#[Time,Distance,Distance,...] label on #-th axis
    floats mag=                  spike magnitudes [nsp]
    int    n#=                    size of #-th axis
    int    nsp=1                 Number of spikes
    float  o#[0,0,...]           origin on #-th axis
    floats p#[0,...]            spike inclination (in samples) [nsp]
    string title=               title for plots
    string unit#[s,km,km,...]    unit on #-th axis
USED IN
    bei/conj/causint
    bei/dpmv/matt
    bei/dpmv/yalei
```

# Madagascar Basic Usage

## ➤ Interacting with files from the command line

### ◆ Get header information by using “sfin”

Demo:

**sfin file1.rsff** or **sfin < file1.rsff**

```
file1.rsff:
in="/RSFDATA/file1.rsff@"
esize=4 type=float form=native
n1=100          d1=0.01          o1=0          label1="Time" unit1="s"
n2=50           d2=10            o2=1000       label2="Distance" unit2="km"

5000 elements 20000 bytes
```

### ◆ Get file attributes by using “sfattr”

Demo:

**sfattr < file1.rsff**

```
*****
rms = 1
mean = 1
2-norm = 70.7107
variance = 0
std dev = 0
max = 1 at 1 1
min = 1 at 1 1
nonzero samples = 5000
total samples = 5000
*****
```

### ◆ Print out data values by using “sfdisfil”

Try:

**sfdisfil < file1.rsff**

Print out data values.		
<b>sfdisfil &lt; in.rsff number=y col=0 format= header= trailer=</b>		
Alternatively, use sfdd and convert to ASCII form.		
<i>int</i>	<b>col=0</b>	Number of columns. The default depends on the data type: 10 for int and char, 5 for float, 3 for complex
<i>string</i>	<b>format=</b>	Format for numbers (printf-style). The default depends on the data type: "%4d " for int and char, "%13.4g" for float, "%10.4g,%10.4gi" for complex
<i>string</i>	<b>header=</b>	Optional header string to output before data
<i>bool</i>	<b>number=y</b> [y/n]	If number the elements
<i>string</i>	<b>trailer=</b>	Optional trailer string to output after data

# Madagascar Basic Usage

Packing header and data

```
[< in.rsfsfprog [> out.rsfsf] out=stdout
```

## ➤ Program execution

- ◆ Single input, single output

```
< input.rsfsfprog [par1=] [par2=] [...] [> output.rsfsf]
```

**Demo:**

```
< file1.rsfsfwindow n2=25 > file2.rsfsf
```

- ◆ Multiple inputs, multiple outputs

```
< input1.rsfsfprog label1=input2.rsfsf label2=output2.rsfsf ... > output1.rsfsf
```

**Example:**

```
< wavelet.rsfsfafmod vel=velocity.rsfsf den=density.rsfsf sou=sources.rsfsf  
rec=receivers.rsfsf wfl=wavefield.rsfsf > data.rsfsf
```

## ➤ Pipes

- ◆ Stdout from one program becomes stdin for the next
- ◆ No intermediate files saved
- ◆ No limit to the number of statements

**Demo:**

```
< file1.rsfsfwindow n2=25 | sftransp > file3.rsfsf
```

**Try:**

```
sfm file1.rsfsf file2.rsfsf file3.rsfsf
```

# Madagascar Basic Usage

## ➤ Plotting

### ◆ Creating plots

VPLOT filters (from \*.rsf to \*.vpl)

Run "**sfdoc stdplot**" for more parameters

List of available plotting programs in Madagascar.	
<b>sfbox</b>	make box-line plots
<b>sfcontour</b>	make contour plots
<b>sfcontour3</b>	make contour plots of 3D surfaces
<b>sfdots</b>	plot signal with lollipops
<b>sfgraph</b>	create line plots, or scatter plots
<b>sfgraph3</b>	generate 3-D cube plots for surfaces.
<b>sfgrey</b>	create raster plots or 2D image plots
<b>sfgrey3</b>	create 3D image plots of panels (or slices) of a 3D cube
<b>sfgrey4</b>	generate movies of 3-D cube plots
<b>sfplotrays</b>	make plots of rays
<b>sfthplot</b>	make hidden-line surface plots
<b>sfwiggle</b>	plot data with wiggly traces

### ◆ Visualizing plots

Pens

"**sfpen**" picks the best pen available

Pen name	Dependency	Output
xtpen	X11	Screen
oglopen	OpenGL	Screen
jpegpen	libjpeg	JPEG
tiffpen	libtiff	TIFF
ppmpen	NetPBM	PPM
gdpen	LibGD, FFmpeg	PNG,JPEG,GIF,MPEG
pngpen	Cairo	PNG
svgpen	Cairo	SVG
pdfpen	Cairo	PDF
pspen	None	Postscript
vppen	None	Vplot filter

### ◆ Format conversion

Supported formats: "avi, eps, gif, jpeg, jpg, mpeg, mpg, pdf, png, ppm, ps, svg, tif, tiff, vpl" based on **appropriate third-party libraries.**

**Example:**

**vpconvert [format=] file.vpl**



# Madagascar Basic Usage

## 3. Python scripts using SCons:

Madagascar's data-processing SConstruct

```
from rsf.proj import * # this statement comes first... ALWAYS.
```

```
Fetch('filename', 'dirname')
```

- A rule for downloading files from a server

```
Flow('target', 'source', 'command')
```

- A rule for making target from source

```
Plot('target', 'source', 'command')
```

- Like Flow but generates a figure file

```
Result('target', 'source', 'command')
```

- Like Plot but generates a final result

```
End() # this always ends the script.
```

- ◆ `scons view / scons file.view`
- ◆ `scons file1.rsf file2.rsf file3.rsf ...`
- ◆ `scons -c`
- ◆ `scons -n`
- ◆ `scons file.lock; scons file.flip`

Tips:

- Suffixes “.rsf”, “.vpl” and prefix “sf” can be dropped in SConstruct.
- You can drop any one if “source” and “target” have the same name only in **Plot** and **Result**.

### Evolution of Build Systems

- **Make (1977)**
  - “Sendmail and make are two well known programs that are pretty widely regarded as originally being *debugged into existence*. That's why their command languages are so poorly thought out and difficult to learn. It's not just you - everyone finds them troublesome.” P. van der Linden
- **Cake (1987)**
- **GNU Make (1988)**
- **SCons (2000)**
- ...



## What is SCons?

- Build system (**Software Construction**)
- Written in **Python**
  - Configuration (**SConstruct** files) are Python scripts
- Built-in support for different languages
- Dependency analysis
- Parallel builds
- Cross-platform
- ...



Steven Knight

Courtesy of Sergey Fomel (School, Beijing, 2011)

# Madagascar Basic Usage

## 3. Python scripts using SConstructs:

### Creating an SConstruct

```
from rsf.proj import *
```

Basic methods of an rsf.proj object.

<code>Fetch(data_file, dir[, ftp_server_info])</code>
A rule to download <data_file> from a specific directory <dir> of an FTP server
<code>Flow(target[s], source[s], command[s][, stdin][, stdout])</code>
A rule to generate <target[s]> from <source[s]> using <command[s]>
<code>Plot(intermediate_plot[, source], plot_command)</code> or <code>Plot(intermediate_plot, intermediate_plots, combination)</code>
A rule to generate <intermediate_plot> in the working directory.
<code>Result(plot[, source], plot_command)</code> or <code>Result(plot, intermediate_plots, combination)</code>
A rule to generate a final <plot> in the special Fig folder of the working directory.

`End()`

- Combinations**
- Movie
  - Overlay
  - SideBySideAniso
  - OverUnderAniso
  - SideBySideIso
  - OverUnderIso
  - TwoRows
  - TwoColumns

### Demo:

```
from rsf.proj import *  
  
# Download data  
Fetch('midpts.hh', 'midpts')  
  
# Select a CMP gather, mute  
Flow('cmp', 'midpts.hh',  
    '''  
    window n3=1 | dd form=native |  
    mutter half=n v0=1.5 |  
    put label=Time unit1=s label2=Offset unit2=km  
    ''')  
Plot('cmp', 'grey title="Common Midpoint Gather" ')  
  
# Velocity scan  
Flow('vscan', 'cmp',  
    'vscan half=n v0=1.4 nv=60 dv=0.02 semblance=n')  
Plot('vscan', 'grey color=j allpos=y title="Velocity Scan" ')  
  
Flow('pick', 'vscan', 'scale axis=2 | pick rect1=10 rect2=10 | window')  
Plot('pick',  
    '''  
    graph transp=y yreverse=y min2=1.4 max2=2.6 wanttitle=n  
    label= unit1= label2= unit2= wantaxis1=n wantaxis2=n pad=n  
    ''')  
Result('pick', 'vscan pick', 'Overlay')  
  
End()
```

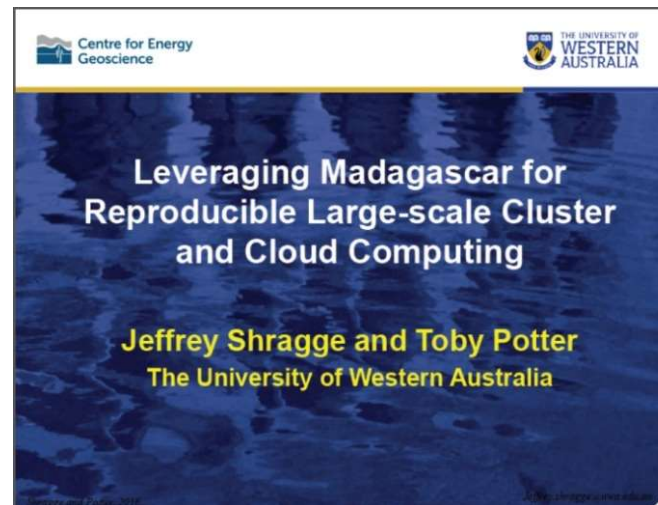
try different one

# Madagascar Basic Usage

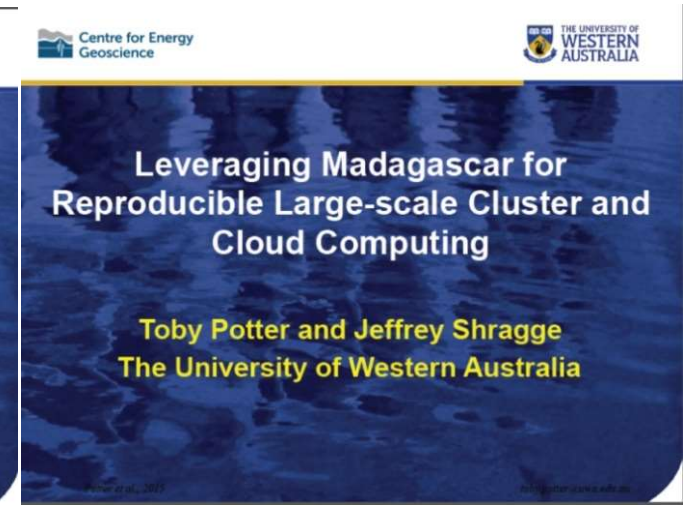
## 4. Parallel computing:

[http://ahay.org/wiki/Parallel\\_Computing](http://ahay.org/wiki/Parallel_Computing)

- ◆ OpenMP
- ◆ MPI
- ◆ OpenMP+MPI
- ◆ pscons



(Shragge and Potter, School, 2016)



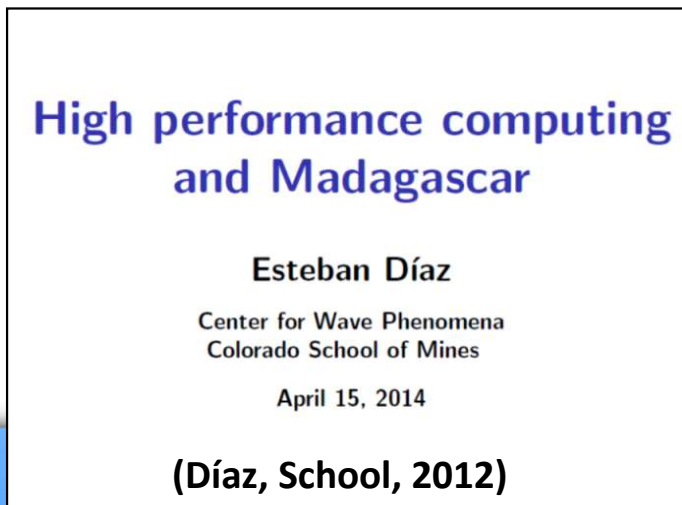
(Potter and Shragge, PASC, 2016)



(Fomel, PASC, 2016)



(Wang, School, 2017)

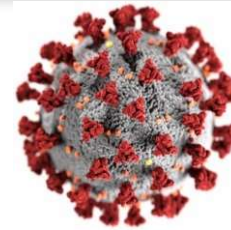




# Outline

- Science and Reproducibility
- Why Madagascar?
- Overview of Madagascar Software Project
- Madagascar Basic Usage
- **Plans for Future Development**
- Open-source Geophysics in China

# Future plans:



## ➤ Next Workshop and School?

COVID-19 makes us to find a new way: virtual meeting -- from anywhere and on any device. Next workshop will be hold on December this year.

## ➤ Fonts supported

Need other fonts supported in plotting, e.g., Chinese font Hershey fonts for Unicode and UTF-8?



## ➤ Test and support new OS

e.g., RHEL 8 / CentOS 8?



## ➤ New servers in China



Madagascar Software Project



# Outline

- Science and Reproducibility
- Why Madagascar?
- Overview of Madagascar Software Project
- Madagascar Basic Usage
- Plans for Future Development
- **Open-source Geophysics in China**

# Open-source Geophysics in China:

➤ More contributions



➤ About GPL license: Free Software not Freeware

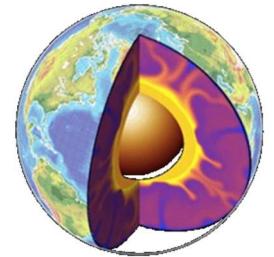


➤ Gitee 码云

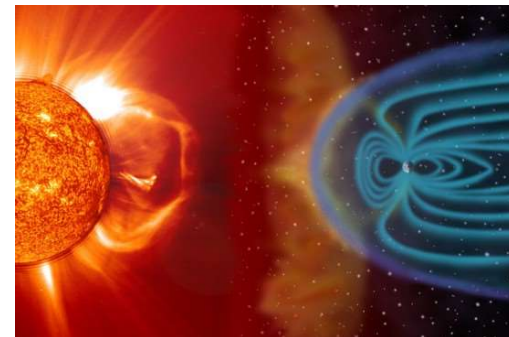
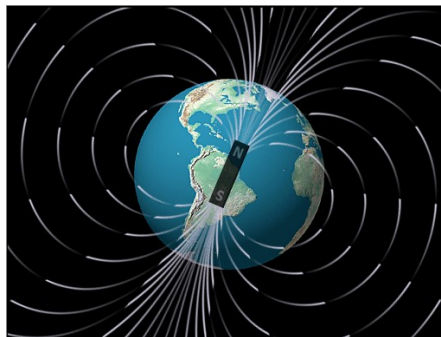
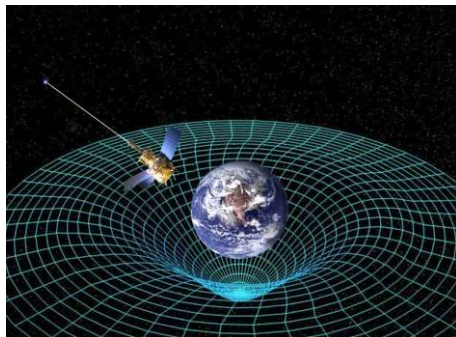


<https://gitee.com/m8r/src>

➤ From Exploration Geophysics to Solid-Earth Geophysics



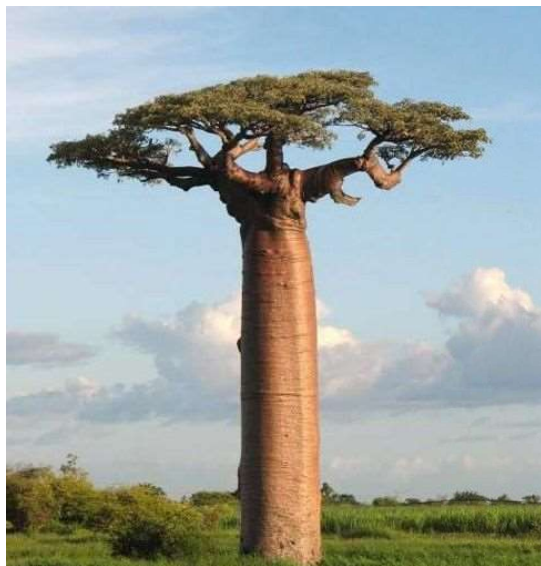
➤ More non-seismic applications, codes, examples, and data



# THANKS FOR ATTENTION AND HAVE A FUN!

- **Madagascar Objectives**

- To make computational research efficient
- To make it easy to share computational results
- To promote an open community



**baobab**  
(*Adansonia digitata* L.)



In real open source, you have the right to control your own destiny.

-- Linus Torvalds



We are here!

