EuroVis Workshop on Visual Analytics (2016)
N. Andrienko and M. Sedlmair (Editors)

MultiLayerMatrix: Visualizing Large Taxonomic Datasets

T. N. Dang[1], H. Cui[2], and A. G. Forbes[1]

[1]University of Illinois at Chicago [2]University of Arizona

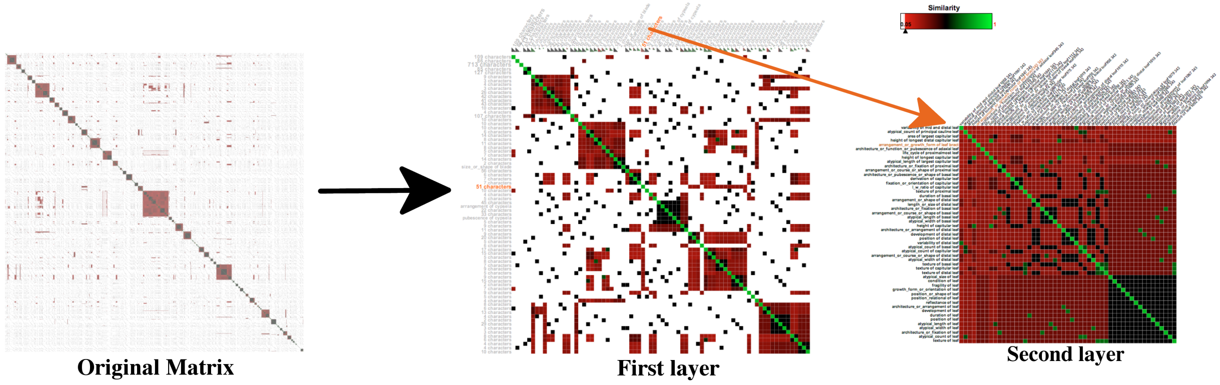


Figure 1: *Visualizing 2048 nodes in a regular adjacency matrix (left) and in a MultiLayerMatrix of two layers: The middle panel shows the first layer, and the right panel shows an example of the second layer, which is shown when users select a cluster in the first layer. Green indicates similar characters while red highlights dissimilarity.*

Abstract

Adjacency matrices can be a useful way to visualize dense networks in which each node is connected to most or all of the rest. However, this technique does not scale well with network size due to limited screen space, especially when the number of rows and columns exceeds the pixel height and width of the screen. We introduce a new scalable technique, MultiLayerMatrix, to visualize very large matrices by breaking them into multiple layers. In our technique, the top layer shows the relationships between different groups of clustered data while each sub-layer shows the relationships between nodes in each group as needed. This process can be applied iteratively to create multiple sub-layers for very large datasets. We illustrate the usefulness of MultiLayerMatrix by applying it to a network representing similarity measures between 2,048 characters in the Asteraceae taxonomy, a rich dataset that describes characteristics of species of flowering plants. We also discuss the scalability of our technique by investigating its effectiveness on a large synthetic dataset with 20,000 columns by 20,000 rows that is initially clustered into 50 distinct groups, and that can then be interactively investigated to examine a further level of detail within a selected cluster.

1. Introduction

Taxon-character matrices are one of the primary tools that biologists traditionally create by hand to classify organisms and to study evolution. With the ongoing development of productivity and text mining software [OK11, RCH*14], it has become possible to create matrices much larger than a manual workflow could support. For example, O'Leary et al. [OBF*13] use a mammal matrix with 86 rows and 4,541 columns, Dececchi et al. [DBLM15] use a matrix with 1,051 rows and 639 columns. Another matrix that was generated using the *ETC Toolkit*[†] — representing about one third of the Asteraceae family — has a size of 978 rows by 2,048

[†] http://etc.cs.umb.edu/etcsite/

columns. The size of these matrices demands novel visualization techniques that are scalable and intuitive to facilitate the curation, management, and use of large taxon-character matrices and their derivatives (e.g. character-character matrices). We were invited to the *Information Visualization of Characters and Taxonomies Workshop* hosted by the *Explorer of Taxon Concept Project* (hereafter, ETC) in May 2015 where we worked with several experts in biology, ecology, and visualization to develop the novel matrix visualization technique reported in this paper.

An obvious solution for visualizing character-by-character similarity is an adjacency matrix where the color in each cell encodes the similarity of each pair of characters. This is depicted in the left panel of Figure 1. However, this technique does not scale well due to the size constraints of a typical computer screen (i.e., there are not enough pixels to represent thousands of characters on each side of a matrix). To account for this scalability constraint, we can provide a high-level abstraction [Zei97] of the original matrix. In other words, instead of drawing every single cell, we can apply a smoothing function on the matrix to ease perceptual recognition [LAE*12]. This technique hides certain details of the original matrix at higher levels, while allowing a user to view details at lower levels through interaction.

In this paper, we introduce a new technique for visualizing large matrices with thousands of items on each dimension. Our technique "breaks" the original matrix into multiple layers by using the leader algorithm [Har75]. The top layer shows the similarity between clusters represented by the leaders. The finest layer shows similarity between characters in each cluster and sub-cluster. In Section 4 we demonstrate how our technique effectively facilitates the exploration of the Asteraceae dataset, which has 2,048 characters and 978 taxa.

The proposed technique aims to achieve the following goals related to the analysis of taxonomies. These design goals are further broken down into specific tasks presented in Section 3.

- **Pattern Discovery and Hypothesis Generation:** An effective visualization should be able to support the discovery of interesting patterns in existing data which could lead to the generation of novel hypotheses. For example, taxonomists, ecologists, and phylogeneticists would like to identify unusual distribution patterns of characters across taxa such as when taxa sharing the same characters are located far apart in a tree.
- **Curation and Management of Existing taxon-by-character data:** Analysts who regularly interact with taxonomies and ontologies have a common need to perform simple curation and editing of existing datasets, such as merging sets of characters and removing characters that are unnecessary or redundant.

2. Related Work

A heat map is a 2D graphical representation of values in a data matrix where cells are color-encoded by the given values. Along the sides of a heat map, additional information can be displayed, such as the dendrogram produced by hierarchical clustering of rows or columns [WF09]. Dececchi et al. [DBLM15] present taxon-by-phenotype matrix heatmaps, where cell colors reflect the number of character states for each anatomical entity for each taxon.

ZAME [EDG*08] visualizes large graphs by aggregating information. Aggregates are arranged into a pyramid hierarchy that allows for on-demand paging to GPU shader programs to support smooth multiscale browsing. In particular, every level of detail has half the number of nodes as the level below it. Consequently, each cell in a higher level is the summary of four cells at the level below it. *ZAME* also supports the rendering glyphs for aggregated cells, which can take various forms, such as histograms, to represent various aggregations of underlying data. *Net-Ray* [KLKF14] projects a large matrix into a smaller one, where an element of the small matrix is set to the number of nonzeros in the corresponding submatrix of the big matrix. This leads to another challenge: small matrix is almost full in most cases. *Net-Ray* handles this problem by reordering nodes in the matrix before projecting and by scaling the x and y axes, as well as the numerical value of each submatrix using different log scales.

The basic difference between *ZAME*, *Net-Ray*, and *MultiLayerMatrix* is in the computation and representation of aggregations. *ZAME* simply groups two neighboring nodes into one in the next abstraction level. *Net-Ray* projects large matrices into a predefined resolution (for example, 1000 by 1000); each cell in the target matrix is given a color based on the average value, thereby giving a false impression about the original matrix. *MultiLayerMatrix* uses the leader algorithm to cluster similar nodes. In particular, two nodes are considered to be similar if they have similar connections to other nodes. For example, in social networks, two people are considered to be similar if they have similar sets of friends. Nodes in a cluster can be from different spatial locations, and cluster size can vary. This algorithm has been successfully used in clustering similar scatterplots in a scatterplot matrix [DW14] and in grouping proteins with similar biologial interactions in a pathway [DMF15].

Some existing work using the hierarchical structure to collapse or expand groups for large adjacency matrix visualization can be found in this state-of-the art report [VBW15]. These techniques are not applicable when there is no inherent hierarchical information attached to the nodeset. In contrast, *MultiLayerMatrix* collapses the characters (nodes) based on the information available within the raw adjacency matrix. There are also several previous works that use an interactive navigable matrix of a previously clustered dataset [AvH04, vH03, AK02].

Henry et al. [HFM07] integrate node-link diagrams and adjacency matrix-based representations into a hybrid visualization, *NodeTrix*. This hybrid representation is suitable for a network where the connections are dense within communities (represented by adjacency matrices), while the connections between these communities are sparse (represented by node-link diagrams). Social networks are an example of such data. For our character similarity data, this technique is unsuitable since the entire network is very dense. Each matrix cell is only empty if we do not have any measure of similarity between two characters, which is rarely the case.

In general, node-link diagrams and other variances of adjacency matrices, such as *Compressed Adjacency Matrices* [DWvW12], *BioFabric* [Lon12], GeneaQuilts [BDF*10], and *DAGView* [KT13], are not suitable for visualizing very dense networks where the degree of nodes is consistently high.

3. Overview of Visualization Tasks

In this section, we provide an overview of some of the main challenges in the visualization of character matrices with thousands of rows and columns.

Taxonomists, ecologists, and phylogeneticists regularly interact with biological taxonomies. They have a common need to cluster related characters and to perform simple editing on the taxonomic data. To this end, a visual analytics platform should allow a user to:

- **T1:** Automatically cluster related characters and provide a high level overview of the large character-by-character table. Users should be able to drill down on the details of these clusters if needed.
- **T2:** Merge sets of characters that are determined by the analyst to be identical for the current analysis.
- **T3:** Separate a selected set of characters from a group that are determined by the analyst to be irrelevant. Moreover, analysts should be able to remove characters that are unnecessary or redundant.

The input data in a typical taxonomic analysis contains both a character-by-character similarity table and a taxon-by-character table, and it is often interesting (albeit challenging) to link both tables to visualize interesting patterns. This could lead to the generation of novel hypotheses. Visualization tasks related to pattern discovery and hypothesis generation include:

- **T4:** Locating potentially important characters as well as missing or redundant characters.
- **T5:** Identify the characters that define or relate to particular sets of taxa within the input taxonomy.
- **T6:** Explore distributions of characters within the taxonomy.

To facilitate these visualization tasks, we propose a new visualization technique which presents a large adjacency matrix in multiple abstraction levels.

4. Our technique

4.1. Input data

The input data provided by the taxonomists in our team contains two tables. The first table is a 2,048 by 2,048 character similarity table. Each cell in this table receives a value in the range of 0 to 1. A value of 1 means two corresponding characters are identical, and they are encoded in green in our visualization. A value of 0 indicates corresponding characters are dissimilar, and they are encoded in red. In some cases we do not have the similarity measures between two characters, and in this case the associated cell in the *MultiLayerMatrix* is left empty. Users can select different color scales (including colorblind safe scales) to encode similarity between characters. In the examples in this paper, we use a *red-green bipolar* color scale since it clearly distinguishes similar and dissimilar characters.

The second table given in the input data is a 978 by 2,048 taxon-by-character table. Each row in this table is a taxon, which contains taxonomic information (i.e. family, tribe, genus, and species), authority information (i.e. authors and publication date), and character values (values on 2,048 characters). This table is very sparse because many characters are unique to a particular taxon or group, or many characters are not described. A visual analytics platform should allow analysts to not only perform curation and management on individual tables but also to link the two tables to highlight interesting distribution patterns.

4.2. Computing the MultiLayerMatrix Visualization

MultiLayerMatrix breaks the input character-by-character matrix into multiple levels using the leader algorithm [Har75]. Given a set of characters and a threshold r, the radius around a cluster's center, the leader algorithm quickly generates a number of clusters and a set of leader characters (**T1**). Each leader represents a cluster of characters.

The assignment of characters to clusters is similar to the k-means algorithm [Har75], but the computational complexity of the leader algorithm is roughly linear (considerably less than that of k-means). The second difference is that we do not need to specify how many clusters that we are looking for (as in k-means). Instead, we want to limit the number of clusters from $\sqrt{n}$ to $2 * \sqrt{n}$ where n is the number of characters. For example, given data with 2048 characters, we expect from 50 to 100 leader characters, and most clusters have fewer than 100 characters. For a larger dataset of 1,000,000 characters, we expect 1,000 clusters, and each clusters will have roughly 1,000 characters. For the same data, if we want to obtain a 3-layer matrix (leader algorithm is computed twice: one for the first layer and one for second layer), we should expect 100 clusters in the top layer, an estimate of 100 sub-clusters in each cluster in the second layer, and an estimate of 100 characters in each sub-cluster in the third layer.

The middle panel of Figure 1 shows a similarity matrix of the 76 clusters of the left panel. When users roll over the cluster name, its details (the second layer matrix of 51 characters) are displayed, as depicted in the right panel of Figure 1. Notice that characters in each cluster are also ordered by their similarities.

MultiLayerMatrix also supports lensing over the matrix to interactively distort the matrix to see more detail around the current mouse position. Figure 2 shows an example. The thumbnails underneath cluster names show a summary of the similarity matrices in the next level. In the lensing area, we can also see that a few names are grayed out. These are distinct characters (without grouping) where similar characters could not be found based on the threshold set by the slider. In brief, the leader algorithm not only groups similar characters into the same clusters but also helps to highlight outlier characters which do not fit into any clusters (**T4**). *MultiLayerMatrix* also supports filter similarity (only plot cells with high similar scores) by using the slider on the top right corner.

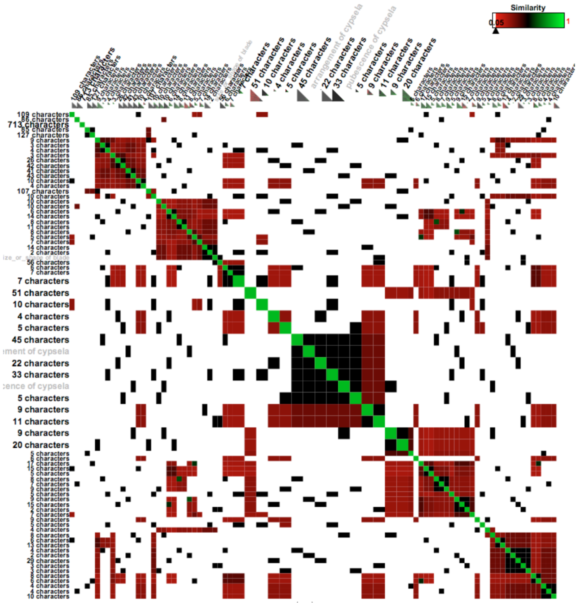


Figure 2: *Visualizing character by character table in the the Asteraceae dataset in the first layer of MultiLayerMatrix. Lensing is applied on the middle section of the matrix.*

4.3. Curation and Management of character clusters

Important visualization tasks supported in *MultiLayerMatrix* include allowing analysts to merge sets of characters that are determined to be identical in a taxonomy (**T2**) and to split a selected set of characters from a group that are determined to be irrelevant (**T3**). This helps to improve the data quality of the matrix. When merging or splitting clusters of characters into one, leaders are recomputed for the new clusters. The leader character is the one which has minimum distance (or most similar) to other characters in the cluster. To see these cluster curations in action, we advise the readers to view the accompanying video available on our GitHub project repository: `https://github.com/CreativeCodingLab/MultiLayerMatrix`.

4.4. Pattern discovery and hypothesis generation

Given one taxonomy with associated characters, analysts would like to zoom into or highlight the branches with certain characters. This feature is particularly interesting to educators and can be used in museums or classrooms as a teaching tool.

MultiLayerMatrix allows users to select a particular branch in the taxonomy and display related characters (**T5**). The related characters are defined as the characters which contain some data in the taxon-by-character table within the selected branch, such as a tribe, a genus, and a species. Figure 3 use the Asteraceae family data. This family contains 10 tribes (in the first column), 137 genera (in the second column), and 537 species (in the third column). The links in this taxonomy are color-encoded by tribe. Ten colors (for ten tribes) were selected from ColorBrewer [HB03]. The thickness of the links are relative to the number of taxa belonging to these branches. Genera (second column) and species (last column) are ordered based on the tribes that they belong to.

In particular, Figure 3 shows an example of selecting a particular species, Californica. As depicted, the Californica species belongs to 4 different genera (Artemisia, Malacothrix, Rafinesquia, and Trixis) which come from 3 different tribes (Anthemideae, Cichorieae, and Mutisieae). Taxonomic names in biology can be complex. At some rank, for example, family, one word name is enough. At sub-ranks, such as tribe or species (sub-species, variety etc.), a binomial naming system is used. For example, a species name has two parts: its genus and its specific epithet. It is not unusual for a specific epithet to be shared by many genera. The naming system's complexity is reflected by the crossing edges between the second and the last column of Figure 3(a). Related characters of the selected species in Figure 3(b) can be displayed (in the form of a smaller similarity matrix) on demand.

T6 requires exploring the distributions of characters within the input taxonomy. In particular, analysts would like to view character distribution patterns across taxa to identify unusual patterns, such as taxa sharing the same characters that are located far apart in a tree. Analysts can select a group of characters in one of the following ways: (1) Characters from a cluster (or multiple clusters) produced by the leader

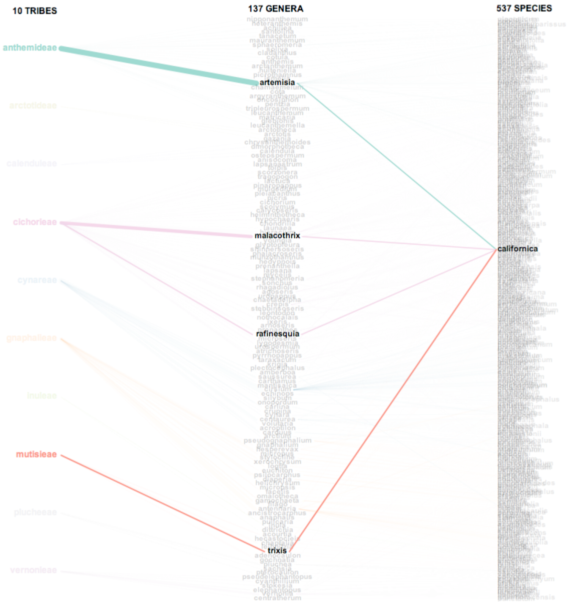


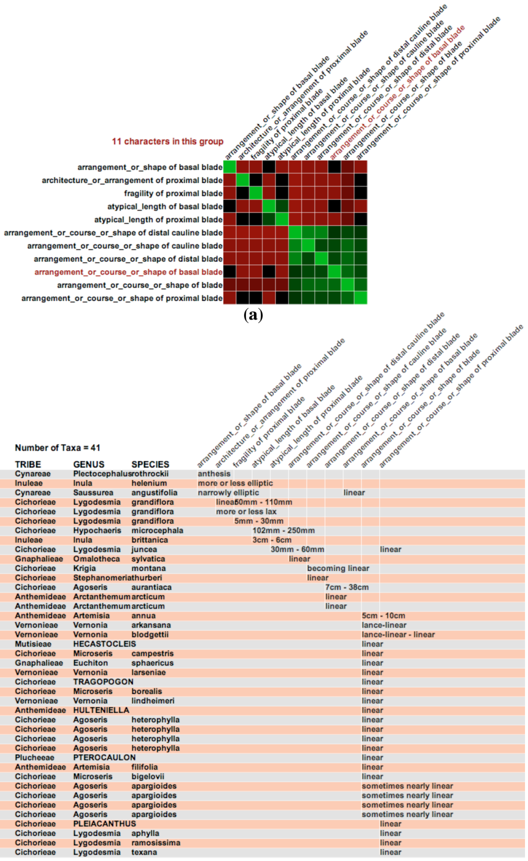
Figure 3: *Visualizing the Asteraceae family which contains 10 tribes (color-encoded), 137 genera, and 537 species: Selecting the Californica species in the last column.*

algorithm presented in Section 4.2 (2) Using rectangular selections to highlight characters of interest. Figure 4(a) shows an example of a selected group of 11 characters. The taxon-by-character table (only characters containing data within the selected taxa) of this group are displayed at the bottom in different orderings. In particular in Figure 4(b), we order taxa by the characters that they first are associated with in the data. This makes ordered indentations on the character columns and helps readability. *MultiLayerMatrix* also supports ordering taxa alphabetically by tribe, genera, and then species. This reveals that taxa sharing the same characters are located far apart in the input taxonomy (**T6**).

5. Scalability

In this section, we explore how well our technique scales to synthetic datasets with over 20,000 elements. This is ten times larger than the number of characters in the example Asteraceae data, so the adjacency matrix size is 100 times larger. Each cell in the 20,000 by 20,000 matrix randomly receives a value from 0 to 1. This is also the largest matrix that can fit into the memory of our testing computer. The test was performed on a 2.5 GHz Intel Core i7, Mac OS X Version 10.10.2, 16 GB RAM running Java 1.7 and Processing 1.5.1. The total running time of the leader algorithm on this synthetic data is about 16 seconds, which generates 50 clusters in the first layer (each cluster contains roughly 400 elements).

This process is completely parallelizable when more resources are available. We propose to take a bottom-up approach in parallelization. For example, the number of characters can be divided evenly to the available processes *m*. Each process will then generate a set of clusters (and leaders) by running the leader algorithm. The results of all processes can then be combined by running the leader algorithm on all leaders (instead of characters) provided by each machine. This would significantly reduce running time.


11 characters in this group
arrangement_or_shape of basal blade
architecture_or_arrangement of proximal blade
fragility of proximal blade
atypical_length of basal blade
atypical_length of proximal blade
arrangement_or_course_or_shape of distal cauline blade
arrangement_or_course_or_shape of cauline blade
arrangement_or_course_or_shape of distal blade
arrangement_or_course_or_shape of basal blade
arrangement_or_course_or_shape of blade
arrangement_or_course_or_shape of proximal blade

(a)

Number of Taxa = 41

FAMILY	TRIBE	GENUS	SPECIES	arrangement_or_shape of basal blade	architecture_or_arrangement of proximal blade	fragility of proximal blade	atypical_length of basal blade	atypical_length of proximal blade	arrangement_or_course_or_shape of distal cauline blade	arrangement_or_course_or_shape of cauline blade	arrangement_or_course_or_shape of distal blade	arrangement_or_course_or_shape of basal blade	arrangement_or_course_or_shape of blade	arrangement_or_course_or_shape of proximal blade
Asteraceae	Cynareae	Plectocephalus	rothrockii	anthesis										
Asteraceae	Inuleae	Inula	helenium	more or less elliptic										
Asteraceae	Cynareae	Saussurea	angustifolia	narrowly elliptic								linear		
Asteraceae	Cichorieae	Lygodesmia	grandiflora		linear	50mm - 110mm								
Asteraceae	Cichorieae	Lygodesmia	grandiflora		more or less lax									
Asteraceae	Cichorieae	Lygodesmia	grandiflora			5mm - 30mm								
Asteraceae	Cichorieae	Hypochaeris	microcephala				102mm - 250mm							
Asteraceae	Inuleae	Inula	brittanica				3cm - 6cm							
Asteraceae	Cichorieae	Lygodesmia	juncea					30mm - 60mm						linear
Asteraceae	Gnaphalieae	Omalotheca	sylvatica						linear					
Asteraceae	Cichorieae	Krigia	montana							becoming linear				
Asteraceae	Cichorieae	Stephanomeria	thurberi							linear				
Asteraceae	Cichorieae	Agoseris	aurantiaca								7cm - 38cm			
Asteraceae	Anthemideae	Arctanthemum	arcticum								linear			
Asteraceae	Anthemideae	Arctanthemum	arcticum								linear			
Asteraceae	Anthemideae	Artemisia	annua										5cm - 10cm	
Asteraceae	Vernonieae	Vernonia	arkansana										lance-linear	
Asteraceae	Vernonieae	Vernonia	blodgettii										lance-linear - linear	
Asteraceae	Mutisieae	HECASTOCLEIS											linear	
Asteraceae	Cichorieae	Microseris	campestris										linear	
Asteraceae	Gnaphalieae	Euchiton	sphaericus										linear	
Asteraceae	Vernonieae	Vernonia	larseniae										linear	
Asteraceae	Cichorieae	TRAGOPOGON											linear	
Asteraceae	Cichorieae	Microseris	borealis										linear	
Asteraceae	Vernonieae	Vernonia	lindheimeri										linear	
Asteraceae	Anthemideae	HULTENIELLA											linear	
Asteraceae	Cichorieae	Agoseris	heterophylla										linear	
Asteraceae	Cichorieae	Agoseris	heterophylla										linear	
Asteraceae	Cichorieae	Agoseris	heterophylla										linear	
Asteraceae	Cichorieae	Agoseris	heterophylla										linear	
Asteraceae	Plucheeae	PTEROCAULON											linear	
Asteraceae	Anthemideae	Artemisia	filifolia										linear	
Asteraceae	Cichorieae	Microseris	bigelovii										linear	
Asteraceae	Cichorieae	Agoseris	apargioides										sometimes nearly linear	
Asteraceae	Cichorieae	Agoseris	apargioides										sometimes nearly linear	
Asteraceae	Cichorieae	Agoseris	apargioides										sometimes nearly linear	
Asteraceae	Cichorieae	Agoseris	apargioides										sometimes nearly linear	
Asteraceae	Cichorieae	PLEIACANTHUS												linear
Asteraceae	Cichorieae	Lygodesmia	aphylla											linear
Asteraceae	Cichorieae	Lygodesmia	ramosissima											linear
Asteraceae	Cichorieae	Lygodesmia	texana											linear

(b)

Figure 4: *Selecting a group of 11 characters: (a) Similarity matrix of 11 ordered characters (b) Taxon-by-character table of the selected characters and the related taxa. Taxa are ordered by the characters that they first are associated with.*

6. Conclusion

In this paper, we presented a novel technique for visualizing and interacting with large matrices by breaking them into multiple layers using the leader algorithm described in Section 4.2. The leader algorithm is roughly linear, making it more scalable for larger networks. We presented this technique using an example dataset which contains a 2,048 x 2,048 character similarity table and a 978 x 2,048 taxon-by-character table. We also ran tests on a 20,000 x 20,000 synthetic character dataset.

The number of nodes (n) in *ZAME* [EDG*08] is reduced by a factor of two after each abstraction level ($n/2$). In *MultiLayerMatrix*, the number of nodes (n) is reduced by a square root factor ($\sqrt{n}$). Therefore, to represent a matrix with potentially millions of rows or columns, we only need to break it into two layers. The first layer displays one thousand by one thousand summary matrix and the second layer displays roughly thousand-by-thousand character matrix. For a larger matrix, more than two layers can be used.

Acknowledgments

This work was funded by the DARPA Big Mechanism Program under ARO contract WF911NF-14-1-0395.

References

[AK02] Abello J., Korn J.: Mgv: a system for visualizing massive multidigraphs. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (Jan 2002), 21–38. 2

[AvH04] Abello J., van Ham F.: Matrix zoom: A visual interface to semi-external graphs. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on* (2004), pp. 183–190. 2

[BDF*10] Bezerianos A., Dragicevic P., Fekete J. D., Bae J., Watson B.: Geneaquilts: A system for exploring large genealogies. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (Nov 2010), 1073–1081. 3

[DBLM15] Dececchi T. A., Balhoff J. P., Lapp H., Mabee P. M.: Toward synthesizing our knowledge of morphology: Using ontologies and machine reasoning to extract presence/absence evolutionary phenotypes across studies. *Systematic Biology* (2015). 1, 2

[DMF15] Dang T., Murray P., Forbes A. G.: Pathwaymatrix: Visualizing binary relationships between proteins in biological pathways. *BioVis 2015 In press* (2015). 2

[DW14] Dang T. N., Wilkinson L.: Scagexplorer: Exploring scatterplots by their scagnostics. In *Proceedings of the 2014 IEEE Pacific Visualization Symposium* (Washington, DC, USA, 2014), PACIFICVIS '14, IEEE Computer Society, pp. 73–80. 2

[DWvW12] Dinkla K., Westenberg M., van Wijk J.: Compressed adjacency matrices: Untangling gene regulatory networks. *Visualization and Computer Graphics, IEEE Transactions on 18*, 12 (Dec 2012), 2457–2466. 3

[EDG*08] Elmqvist N., Do T.-N., Goodell H., Henry N., Fekete J.: Zame: Interactive large-scale graph visualization. In *Visualization Symposium, 2008. PacificVIS '08. IEEE Pacific* (March 2008), pp. 215–222. 2, 6

[Har75] Hartigan J.: *Clustering Algorithms*. John Wiley & Sons, New York, 1975. 2, 3

[HB03] Harrower M., Brewer C. A.: Colorbrewer.org: An online tool for selecting color schemes for maps. the cartographic. *Journal* (2003), 27–37. 4

[HFM07] Henry N., Fekete J.-D., McGuffin M. J.: Nodetrix: A hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov. 2007), 1302–1309. 2

[KLKF14] Kang U., Lee J.-Y., Koutra D., Faloutsos C.: Net-ray: Visualizing and mining billion-scale graphs. In *Advances in Knowledge Discovery and Data Mining*, Tseng V., Ho T., Zhou Z.-H., Chen A., Kao H.-Y., (Eds.), vol. 8443 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, pp. 348–361. 2

[KT13] Kornaropoulos E. M., Tollis I. G.: Dagview: An approach for visualizing large graphs. In *Proceedings of the 20th International Conference on Graph Drawing* (Berlin, Heidelberg, 2013), GD'12, Springer-Verlag, pp. 499–510. 3

[LAE*12] Lehmann D. J., Albuquerque G., Eisemann M., Magnor M., Theisel H.: Selecting coherent and relevant plots in large scatterplot matrices. *Comp. Graph. Forum 31*, 6 (Sept. 2012), 1895–1908. 2

[Lon12] Longabaugh W.: Combing the hairball with biofabric: a new approach for visualization of large networks. *BMC Bioinformatics 13*, 1 (2012), 275. 3

[OBF*13] O'Leary M. A., Bloch J. I., Flynn J. J., Gaudin T. J., Giallombardo A., Giannini N. P., Goldberg S. L., Kraatz B. P., Luo Z.-X., Meng J., Ni X., Novacek M. J., Perini F. A., Randall Z. S., Rougier G. W., Sargis E. J., Silcox M. T., Simmons N. B., Spaulding M., Velazco P. M., Weksler M., Wible J. R., Cirranello A. L.: The placental mammal ancestor and the postâĂŞk-pg radiation of placentals. *Science 339*, 6120 (2013), 662–667. 1

[OK11] OâĂŹLeary M. A., Kaufman S.: Morphobank: phylophenomics in the âĂIJcloudâĂİ. *Cladistics 27*, 5 (2011), 529–537. 1

[RCH*14] Rodenhausen T., Cui H., Huang F., Ludascher B., Macklin J., Morris B., Yu S.: Etc: From description to matrix and beyond in a web-based toolbox. *TWDG Meeting* (2014). 1

[VBW15] Vehlow C., Beck F., Weiskopf D.: The State of the Art in Visualizing Group Structures in Graphs. In *Eurographics Conference on Visualization (EuroVis) - STARs* (2015), Borgo R., Ganovelli F., Viola I., (Eds.), The Eurographics Association. 2

[vH03] van Ham F.: Using multilevel call matrices in large software projects. *Information Visualization, IEEE Symposium on 0* (2003), 29. 2

[WF09] Wilkinson L., Friendly M.: The history of the cluster heat map. *The American Statistician 63*, 2 (2009), 179–184. 2

[Zei97] Zeitz C. M.: Expertise in context. MIT Press, Cambridge, MA, USA, 1997, ch. Some Concrete Advantages of Abstraction: How Experts' Representations Facilitate Reasoning, pp. 43–65. 2