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]Electronic Visualization Laboratory, University of Illinois at Chicago [2]School of Information, University of Arizona

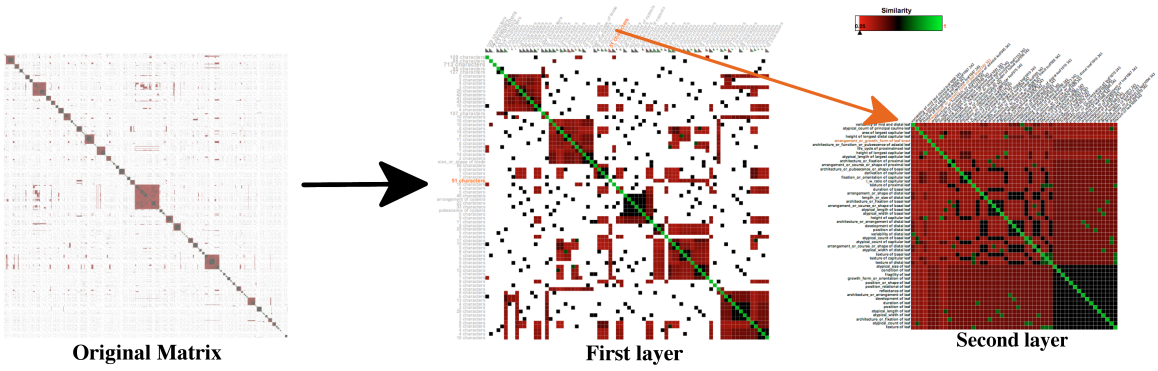

Figure 1: *Visualizing 2048 nodes in a regular adjacency matrix (left) and in a MultiLayerMatrix of two layers: The middle panel shows the first layer, and the right panel shows an example of the second layer, which is shown when users select a cluster in the first layer. Green indicates similar characters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

Abstract

Adjacency matrices can be a useful way to visualize dense networks. However, they do not scale well as the network size increases 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.

1. Introduction

Taxon-character matrices are one of the primary tools that biologists use to classify organisms and to study evolution. Although traditionally created by hand, newer software tools [OK11, RCH*14] make it possible to create matrices much larger than a manual workflow could support. For example, O'Leary et al. [OBF*13] make use of a matrix with 86 rows and 4,541 columns and Dececchi et al. [DBLM15] use a matrix with 1,051 rows and 639 columns. The size of these matrices demands novel visualization techniques that are scalable and intuitive to support the curation, management, and exploration of large taxon-character matrices and their derivatives (e.g. character-character matrices).

An obvious way to visualize character-by-character similarity is by using an adjacency matrix where the color in each cell encodes the similarity of each pair of characters. An example of this approach is depicted in the left panel of Figure 1. However, this approach 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

DOI: 10.2312/eurova.20161125

each side of a matrix). To mitigate this scalability issue, we can provide a high-level abstraction [Zei97] of the original matrix. Rather than drawing every single cell, we can instead apply a smoothing function on the matrix to ease perceptual recognition [LAE*12]. By so doing, we hide certain fine-grained details of the original matrix at higher levels while still enabling a user to interactive view these details on demand.

In this paper we introduce *MultiLayerMatrix*, a new technique for visualizing large matrices with thousands of items. 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 additional layers shows similarity between characters in each cluster and sub-cluster. Our technique aims to achieve the following goals related to the analysis of taxonomies:

- **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 share the same characters but are located far apart in a phylogenetic tree.
- **Curation and management of existing taxon-by-character data:** Analysts who regularly interact with taxonomies and ontologies have a common need to perform curation and editing tasks for existing datasets, such as merging sets of characters and removing characters that are unnecessary or redundant.

These high-level design goals are supported through enabling the specific tasks described in Section 3.

2. Related Work

In general, node-link diagrams and most varieties of adjacency matrices, such as *NodeTrix* [HFM07], *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. To mitigate difficulties in representing dense networks, *ZAME* [EDG*08] visualizes large graphs by aggregating information. Aggregates are arranged into a pyramid hierarchy that allows for on-demand paging to GPU 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. Similarly, *Net-Ray* [KLKF14] projects a large matrix onto a smaller one, where an element of the small matrix is set to the number of non-zero elements in the corresponding submatrix of the big matrix. However, this leads to another challenge: the 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 and the numerical value of each submatrix.

A main difference between *ZAME*, *Net-Ray*, and our technique, *MultiLayerMatrix* is the way in which aggregations are computed and represented. *ZAME* simply groups two neighboring nodes into one element in subsequent abstraction levels. *Net-Ray* projects large matrices into a predefined resolution and each cell in the target matrix is given a color based on the average value, which can present a false impression about the data in 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 thus be drawn from different spatial locations and cluster sizes can vary. This algorithm has been successfully used in clustering similar scatterplots [DW14b], images [DW14a], and proteins with similar biochemical interactions [DMF15].

Existing work that takes advantage of the hierarchical structure to collapse or expand groups for large adjacency matrix visualization is described in a recent state-of-the art report by Vehlow et al. [VBW15]. In contrast, *MultiLayerMatrix* collapses the characters (nodes) based on the data available directly within the raw adjacency matrix, that is, without requiring a specified hierarchical structure. Inspired by previous work [AvH04,AK02,DFLF15,PF15,vH03], our technique also enables the interactive navigation of the matrix layout, as discussed below.

3. Overview of Visualization Tasks

Taxonomists, ecologists, and phylogeneticists regularly need interact with biological taxonomies in order to make sense of data for a range of scientific tasks. They have a common need to cluster related characters and to manage and to edit taxonomic data. To this end, an effective visualization tool should enable a user to:

- **T1:** Automatically cluster related characters and provide a high level overview of the large character-by-character table.
- **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 is determined by the analyst to be irrelevant. Moreover, the analyst 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 in order 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:** Identifying characters that define or relate to particular sets of taxa within the input taxonomy.
- **T6:** Exploring distributions of characters in the taxonomy.

4. The *MultiLayerMatrix* Visualization Technique

4.1. Input Data

The input data (provided by the taxonomists on 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 is encoded using the color green in our visualization. A value of 0 indicates that corresponding characters are dissimilar and is encoded in red. In some cases we do not know the similarity measurement between two characters; in this case the associated cell in the *MultiLayerMatrix* is left empty.

The second table provided 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. This table is very sparse since many characters are unique to a particular taxon or group and because many characters are not fully 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. However, the computational complexity of the leader algorithm is roughly linear (and considerably more efficient than that of k-means). Another difference is that we do not need to specify how many clusters that we are looking for (as is required 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, each of which will contain roughly 1,000 characters. For the same data, if we want to obtain a 3-layer matrix, the leader algorithm is computed twice, once for the first layer and second time for second layer. In this case, we should expect 100 clusters in the top layer, approximately 100 sub-clusters for each cluster in the second layer, and around 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 a user mouses over the cluster name, its details (the second layer matrix of 51 characters) are displayed, as depicted in the right panel of Figure 1.

MultiLayerMatrix supports lensing over the matrix to interactively distort the matrix in order to see more detail around the current mouse position. Figure 2 shows an example of interactively lensing. The thumbnails underneath cluster names show a summary of the similarity matrices available in the next level. In the lensing area, we can also see that a few names are grayed out. These are distinct characters where similar characters could not be found based on a threshold set by the user (via an interactive slider). In brief, the leader algorithm not only groups similar characters into the same clusters but also helps to highlight outlier characters that do not fit into any clusters (**T4**).

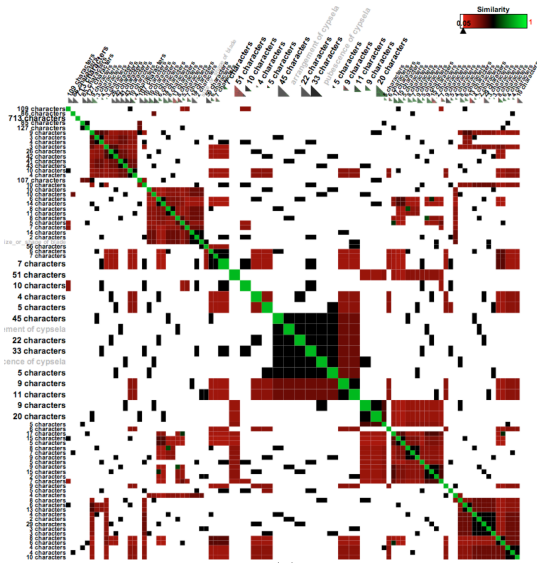

Figure 2: *Visualizing character by character table in the the Asteraceae dataset in the first layer of MultiLayerMatrix.*

4.3. Curation and Management of Character Clusters

Important visualization tasks supported in *MultiLayerMatrix* include merging sets of identical characters (**T2**) and splitting apart characters from a group that are determined to be irrelevant (**T3**), which can help to improve the data quality of the matrix. Leaders are recomputed when merging or splitting clusters of characters. The leader character is the one which has minimum distance (or most similar) to other

characters in the cluster. The supplementary video shows examples of these cluster curations in action. (The video is also available via our project repository at https://github.com/CreativeCodingLab/MultiLayerMatrix).

4.4. Pattern Discovery and Hypothesis Generation

Given a taxonomy with associated characters, analysts would like to zoom into or highlight the branches with certain characters. This feature is 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 taht contain some data in the taxon-by-character table within the selected branch, such as a tribe, a genus, or a species. Figure 3 uses 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 the 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.

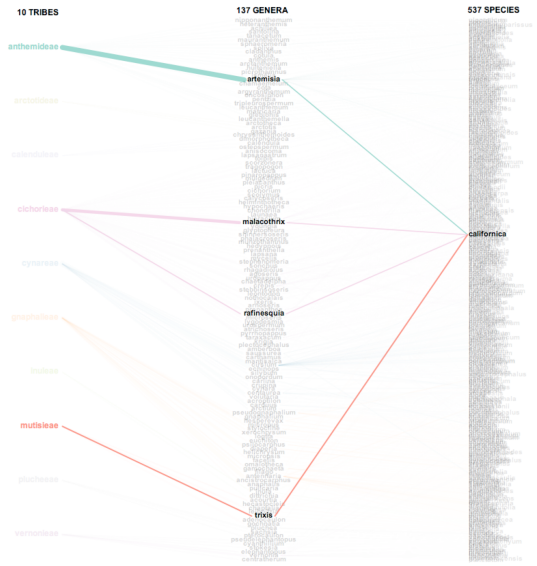

Figure 3: *Visualizing the Asteraceae family which contains 10 tribes (color-encoded), 137 genera, and 537 species: Selecting the Californica species in the last column.*

Figure 3 shows an example where a particular species, Californica, is selected. 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 ranks (for example, family) one word name is sufficient. However, 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 (that is, its common name). 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 in order to identify unusual patterns, such as taxa sharing the same characters that are located far apart in a tree. Analysts can choose a group of characters by selecting characters from a cluster or by using the rectangular selection mode to highlight particular characters of interest.

Our technique effectively 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. This 20,000 x 20,000 matrix requires nearly all of the memory of our testing computer, a 2.5 GHz Intel Core i7 with 16 GB RAM. The total running time of the leader algorithm on this synthetic dataset is close to 16 seconds on average, generating 50 clusters in the first layer (where each cluster contains roughly 400 elements).

5. 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 than other techniques when working with large 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. Future work will explore optimizing our technique (which is completely parallelizable) for even larger datasets. The number of characters can be divided evenly to make use of the available processes and 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, significantly reducing the running time.

Acknowledgments

This work was funded in part by the DARPA Big Mechanism Program under ARO contract WF911NF-14-1-0395 and also by the NSF Advances in Biological Informatics program, award #DBI-1147266.

References

[AK02] ABELLO J., KORN J.: MGV: A system for visualizing massive multidigraphs. *IEEE Trans. on Visualization and Computer Graphics 8*, 1 (2002), 21–38. 2

[AvH04] ABELLO J., VAN HAM F.: Matrix Zoom: A visual interface to semi-external graphs. In *Proc. of IEEE Symp. on Information Visualization* (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 Trans. on Visualization and Computer Graphics 16*, 6 (2010), 1073–1081. 2

[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

[DFLF15] DANG T. N., FRANZ N., LUDÄSCHER B., FORBES A. G.: ProvenanceMatrix: A visualization tool for multi-taxonomy alignments. In *Proc. of the ISWC Workshop on Visualization and User Interfaces for Ontologies and Linked Data (VOILA)* (2015), vol. 1456 of *CEUR Workshop Proceedings*, pp. 13–24. 2

[DMF15] DANG T. N., MURRAY P., FORBES A. G.: PathwayMatrix: Visualizing binary relationships between proteins in biological pathways. *BMC Proceedings 9*, 6 (2015), S3. 2

[DW14a] DANG T. N., WILKINSON L.: *Advances in Visual Computing: 10th International Symposium, ISVC 2014, Las Vegas, NV, USA, December 8-10, 2014, Proceedings, Part II*. Springer International Publishing, Cham, 2014, ch. PixSearcher: Searching Similar Images in Large Image Collections through Pixel Descriptors, pp. 726–735. 2

[DW14b] DANG T. N., WILKINSON L.: ScagExplorer: Exploring scatterplots by their scagnostics. In *Proc. of the IEEE Pacific Visualization Symposium* (2014), pp. 73–80. 2

[DWvW12] DINKLA K., WESTENBERG M., VAN WIJK J.: Compressed adjacency matrices: Untangling gene regulatory networks. *Visualization and Computer Graphics, IEEE Trans. on 18*, 12 (Dec 2012), 2457–2466. 2

[EDG*08] ELMQVIST N., DO T.-N., GOODELL H., HENRY N., FEKETE J.: ZAME: Interactive large-scale graph visualization. In *Proc. of the IEEE Pacific Visualization Symposium* (2008), pp. 215–222. 2

[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 Trans. on Visualization and Computer Graphics 13*, 6 (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 *Proc. of the International Conference on Graph Drawing* (2013), pp. 499–510. 2

[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. 2

[OBF*13] O'LEARY M. A., BLOCH J. I., FLYNN J. J., GAUDIN T. J., GIALLOMBARDO A., GIANNINI N. P., ET AL.: The placental mammal ancestor and the post-K-Pg radiation of placentals. *Science 339*, 6120 (2013), 662–667. 1

[OK11] O'LEARY M. A., KAUFMAN S.: MorphoBank: Phylophenomics in the cloud. *Cladistics 27*, 5 (2011), 529–537. 1

[PF15] PADUANO F., FORBES A. G.: Extended LineSets: A visualization technique for the interactive inspection of biological pathways. *BMC Proceedings 9*, 6 (2015), S4. 2

[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. *Biodiversity Information Standards – Taxonomic Databases Working Group* (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). 2

[vH03] VAN HAM F.: Using multilevel call matrices in large software projects. In *Proc. of IEEE Symp. on Information Visualization* (2003), pp. 227–232. 2

[Zei97] ZEITZ C. M.: Some concrete advantages of abstraction: How experts' representations facilitate reasoning. In *Expertise in Context*, Feltovich P. J., Ford K. M., Hoffman R. R., (Eds.). MIT Press, Cambridge, MA, USA, 1997, pp. 43–65. 2