Dendrogramix: a Hybrid Tree-Matrix Visualization Technique to Support Interactive Exploration of Dendrograms

Renaud Blanch*
Univ. Grenoble Alpes, LIG
F-38000 Grenoble, France

Rémy Dautriche†
Univ. Grenoble Alpes, LIG
STMicroelectronics
F-38920 Crolles, France

Gilles Bisson‡
CNRS, LIG
F-38000 Grenoble, France

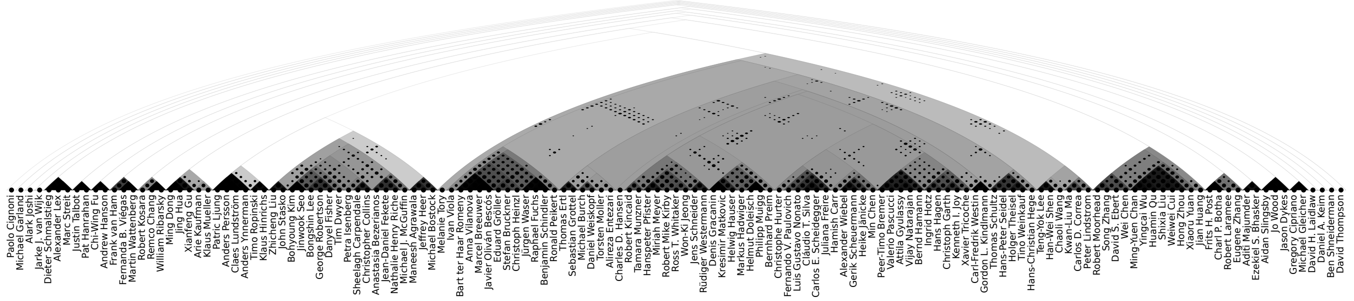


Figure 1: Dendrogramix visualizing 6 years (2006–2011) of co-authorship at the IEEE InfoVis conference.

Abstract

Clustering is often a first step when trying to make sense of a large data set. A wide family of cluster analysis algorithms, namely *hierarchical clustering* algorithms, does not provide a partition of the data set but a hierarchy of clusters organized in a binary tree, known as a *dendrogram*. The dendrogram has a classical node-link representation used by experts for various tasks like: to decide which subtrees are actual clusters (e.g., by cutting the dendrogram at a given depth); to give those clusters a name by inspecting their content; etc. We present *Dendrogramix*, a hybrid tree-matrix interactive visualization of dendrograms that superimposes the relationship between individual objects on to the hierarchy of clusters. *Dendrogramix* enables users to do tasks which involve both clusters and individual objects that are impracticable with the classical representation, like: to explain why a particular objects belongs to a particular cluster; to elicit and understand uncommon patterns (e.g., objects that could have been classified in a totally different cluster); etc. Those sensemaking tasks are supported by a consistent set of interaction techniques that facilitates the exploration of large clustering results.

Keywords: Agglomerative hierarchical clustering, dendrogram, Dendrogramix, hybrid visualization, interaction.

Index Terms: H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces—GUI; I.3.6 [Computer Graphics]: Methodology and Techniques —Interaction techniques

1 Introduction

Agglomerative hierarchical clustering (AHC) [17] is often used to group items into clusters as soon as a similarity metric enables the pairwise comparison of items. AHC is widely used because it is easy for users to figure out how the classification is built, but also because they can choose the number of clusters after the classification has been made. The understanding that users can get of AHC relies heavily on the canonical visualization of the cluster hierarchy resulting from an AHC: the dendrogram (e.g., Figure 2.b).

The dendrogram provides a visualization of the binary tree of the clusters built by the AHC (given by its node-link structure), but also an information about the homogeneity of each cluster (given by the height of the internal nodes). This visual encoding of the cluster's homogeneity helps the user to choose a level at which to cut the hierarchy in order to produce a final partition into actual clusters.

If the dendrogram visualization allows to compare clusters —by their homogeneity (their height), but also by their cardinality (their width)—, the dendrogram discards completely the original information on items, and thus does not allow any comparison involving them. Without this information in the dendrogram, it is not possible to answer some kind of questions, like: why did a specific item ended in a specific cluster, is it because it is very similar to a single other item of the cluster, or is it because it is not very dissimilar with any other member of the cluster?

In this paper, we introduce an alternative to dendrogram, namely *Dendrogramix*, which provides, within the same screen real estate, a visualization of the hierarchy of clusters, of their homogeneity, but also of the similarity between items. This visualization comes with a set of carefully designed interaction techniques which allows users to explore the hierarchy of clusters in order to make sense of the result of the AHC. Figure 1 shows the Dendrogramix resulting from the AHC of authors from six past (2006–2011) InfoVis conferences[1]. The 143 authors (among 1142) that have published at least 3 papers (among 512) over this period are grouped according to the similarity of their set of co-authors[2].

2 Related Work

The clustering itself uses the well-known AHC method [17]. Dendrogramix is not the first attempt at showing together the clustering result and the details about items, but previous works in this area

*e-mail: renaud.blanch@imag.fr

†e-mail: remy.dautriche@imag.fr

‡e-mail: gilles.bisson@imag.fr

[1] The data set has been automatically extracted from on-line digital libraries, and manually curated for author deduplication.

[2] In this example, the measure of similarity is the cosine similarity, and the distance between clusters uses the single-linkage method.

all rely on the juxtaposition of two visualizations: the classical dendrogram and another visualization for the individual items. Dendrogramix is not a juxtaposition of two visualizations but rather a mix of two visualizations. It is thus more close to recent works in the InfoVis community about hybrid visualizations. The interaction techniques provided by our Dendrogramix are also comparable to recent works focused on the interaction with visualizations, especially interaction that exploit the structure of the data visualized to guide the user.

2.1 Hierarchical Clustering Visualization

Several visualization have been proposed to display the information about items together with the clustering result (see Wilkinson & Friendly survey [18] for a comprehensive overview of those techniques). The raw information on items can be shown with a heat map [16] placed side by side with the dendrogram. Such visualizations have been improved with interactions and overview+detail view to cope with the scaling issues encountered when dealing with large data sets [5].

Another way to display information about items is to show their similarity matrix rather than their raw vectors, as proposed by Gower & Digby [7]. It has the advantage of saving space when the item vectors have more dimensions than the number of items, as the similarity matrix is square and symmetric. It has also the advantage of actually showing the similarity between items as seen by the system, rather than letting the user reconstruct it from the visual similarity of the row vectors from the heat map which can be misleading. For those reasons we have also chosen to use the similarity matrix as a starting point to provide information on items.

Our main contribution on the graphic representation is that the Dendrogramix embeds the information about items into the visualization of the tree representing the clustering result.

2.2 Hybrid Visualizations

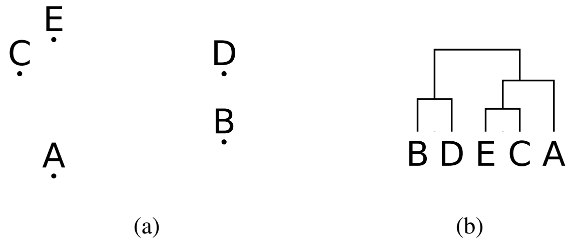
Another way to cope with scaling issues is to use different representations for sub-parts of a data-set (e.g., node-link and treemap for trees, as in elastic hierarchies [19], or node-link and adjacency matrix for graphs, as in MatLink [8], NodeTrix [9], or TreeMatrix [13]). The choice for a specific representation for a part of the data can be made interactively by the user, as in NodeTrix, or take advantage of theoretical results about the space-efficiency of various representations (e.g, for trees [11]) to automatically switch from a representation to another at various levels of aggregation. Javed & Elmqvist proposed a design space of such composite visualizations and some guidelines to choose a relevant composition [10].

Such hybrid visualizations have been used to display large dendrograms. In Stacked Trees [3], the classical dendrogram visualization is used for the largest clusters, but above an homogeneity threshold the visualization switches to a stack of leaves rather than going on with the tree representation. This saves space at the expense of loosing structural information.

Our Dendrogramix is also hybrid but in a different way: it combines the visualizations of two different data sets that do not have the same structure: the input of the AHC —a matrix storing the items similarity—, and the output of the AHC —a binary tree depicting the clusters and their homogeneity.

2.3 Interaction Techniques

The interaction techniques designed to explore the Dendrogramix follow the principles of direct manipulation [15], especially the fact that they should be rapid, reversible, and incremental. The main challenge here is that the data structures manipulated are inherently discrete. Making the interaction continuous requires the use of animated transitions controlled either by the system or by the user to switch between coherent states of the visualization. Zoomable Treemaps [4] is a good example of previous work that provides a continuous interaction with the discrete data structure of a tree visualized as a treemap. We borrowed from Zoomable Treemaps the idea of using a crossing-based interaction [1] to perform an advanced selection (of two clusters simultaneously in our case; of an arbitrary internal node of the treemap in their case).



(a) (b)

Figure 2: Data set used for illustration purpose in Figure 3: (a) points of the plan; and (b) dendrogram of the data using Euclidean distance and single-linkage agglomerative hierarchical clustering.

Most of the interaction techniques proposed by Dendrogramix fit in the framework of interactions with hierarchical aggregation proposed by Elmqvist & Fekete [6]. However, they do not consider interactions altering the layout (such as node reordering) in their framework.

3 Dendrogramix

A Dendrogramix is an hybrid visualization that mixes the similarity matrix of items with the binary tree of clusters resulting from their AHC. We first describe how the visualization is built and what kind of observations can be made using it. We then describe the interaction techniques that allow its exploration.

3.1 Visualization

Figure 2.a shows the data set —five points of the plan— used below to illustrate how a Dendrogramix is built. Figure 2.b shows the result of an AHC of this data set using the Euclidean distance to measure the similarity between two points, and the minimum distance between the elements of two clusters to generalize this similarity measure to clusters (i.e., the single linkage method).

3.1.1 Construction

Using the tree of clusters and the similarity matrix as inputs, the Dendrogramix is built:

1. by encoding the similarity matrix using the size of the circular dots to denote similarity (large dots means similar items) —Figure 3.a;

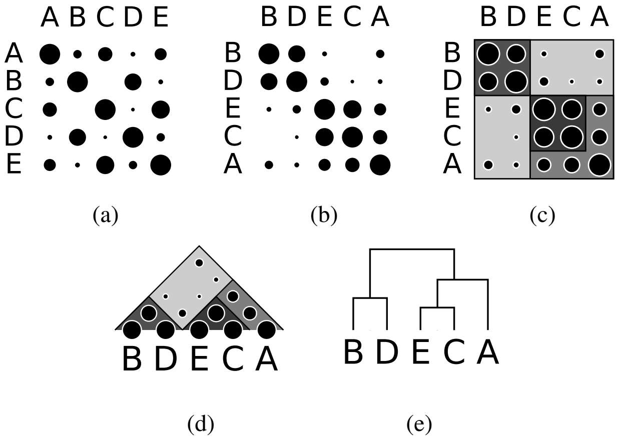
2. by reordering the matrix with an order of the items given by a traversal of the leaves of the clusters tree —Figure 3.b;

3. by highlighting the boundaries of the clusters (which are contiguous because of the previous step), and by encoding their homogeneity using the level of gray of their background (black means an homogeneous cluster) —Figure 3.c; and

4. by keeping only an half of the matrix (since it is symmetrical, no information is lost), and by tilting it at 45° to have the root of the tree at the top and its leaves at the bottom —Figure 3.d.

For the first step, the similarity is normalized, and the radius of the circular dots are equal to the square root of this normalized similarity, so that the area of the dots is proportional to the similarity.

For the second step, we use an optimal order for the leaves, i.e. an order that minimizes the sum of the distances between adjacent items.



(a) (b) (c)

(d) (e)

Figure 3: From similarity matrix to Dendrogramix: (a) graphic encoding of the similarity; (b) matrix reordered with an (optimal) order compatible with a traversal of the tree of clusters; (c) graphic encoding of clusters; (d) Dendrogramix compared to (e) classic dendrogram showing the same hierarchical clustering of the same data.

For the third step, the mapping between the cluster homogeneity and the gray scale is linear: black denotes a cluster as homogeneous as possible (consisting of identical items), and white denotes the less homogeneous cluster (the root cluster consisting of the whole set of items).

For the last step, besides the half-matrix tilt, it is possible to apply a deformation that compresses the matrix. The farther from the diagonal a point is, the more compressed it is. This makes sense because, if the ordering of the leaves is good, then most of the information is concentrated near the diagonal of the matrix. This distortion thus allows to save space while keeping most of the information legible. This space can then be used in turn to show the labels of the items.

3.1.2 Design Rationale

Various choices led to the design presented above. First, the superimposition of the matrix and the tree has been chosen because it allows for a direct comparison of groups homogeneity and item similarity. We could have juxtaposed the similarity matrix to a classical dendrogram, but it would have used more space. Moreover, dendrograms are often juxtaposed with heat maps (that show the raw items' data). With the Dendrogramix, we keep this juxtaposition possible while also showing the similarities.

Once this choice is made, the retinal variables available to encode the items' similarities and the clusters' homogeneities are limited as the positions are already fixed by the tabular layout. The best remaining retinal variables (using Bertin's terminology) are thus size and value. Since the clusters' geometry is defined by their containment, we used their value to encode homogeneity. This makes it hard to do absolute judgments on homogeneity, but the most frequent task is to compare the homogeneity of one cluster to the homogeneity of its parent, and this task maps nicely to a relative judgment on the values (contrast) of two adjacent areas.

The size is then used to encode the similarity between items. As the homogeneity of a cluster is an aggregation of the similarities of its items, the two retinal variables correlate: the value of a cluster is directly linked to the sizes of the similarities. It is then easy to spot the abnormal similarities (large dots on a light background or missing dots on a dark background) that are worth investigating.

3.2 Interaction Techniques

Figure 1 shows a Dendrogramix that displays the AHC of 142 InfoVis authors. Figure 4 shows a close-up on a specific cluster. On this cluster we can make some observations that show how Dendrogramix is effective at helping understand the result of the AHC, and how it is linked to the raw data. The Dendrogramix is interactive: it comes with a coherent set of interaction techniques that allows making such observations through the data set exploration.

3.2.1 Sample Exploration

We can see that authors that work at the same place are effectively grouped together. The first cluster on the left is obviously a research group (the SciVis group at Linköping University). Then we have a larger cluster consisting of people from various groups (the Information Interfaces Group at Georgia Tech, the VibeVis group at Microsoft Research, the Aviz group at INRIA, etc.) that had links at that time through various collaborations.

We can see that those groups are linked by people that are prolific and collaborate a lot (e.g., Nathalie Henry Riche, Bongshin Lee or Michael McGuffin). Those "social" people that link smallest clusters are characterized by their long series of dots on their row/column of the similarity matrix. This kind of information would be totally lost if the clustering result were presented using a classical dendrogram.

The similarity information on items helps understand clusters at a finer level. The cluster on the right, consisting of three people (Maneesh Agrawala, Jeffrey Heer, and Michael Bostock), is very homogeneous. But its links to the other clusters are not: if the three of them are linked with George Robertson, the dots pattern also conveys the information that it is Jeffrey Heer who connects this cluster to the InfoVis authors community.

To draw such conclusions, some interactive exploration is needed. The interaction techniques used are described below.

3.2.2 Items Comparison

Hovering the visualization with the mouse cursor allows to get information about the items (Figure 5).

When hovering a label, the corresponding item is highlighted, as well as its row and column in the matrix (e.g., Figure 5.a shows the highlight of George Robertson). When an item is highlighted, its similarity with the other items is shown above the labels with a row of rectangles, their value giving the similarity.

When hovering the matrix, the position of the cursor designate two items (a row and a column). Their labels, as well as their rows

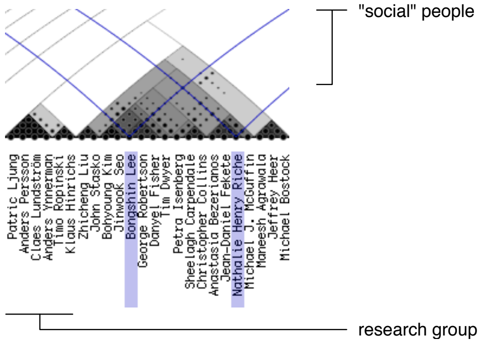
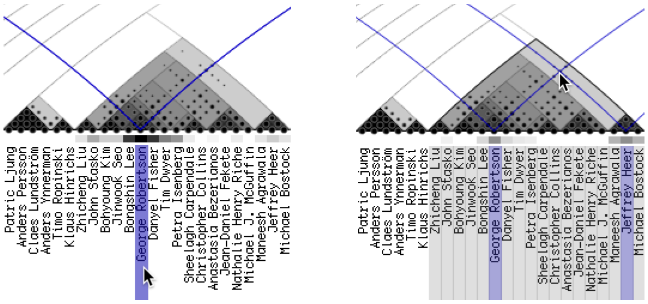


Figure 4: Cluster detail, research groups are visible as homogeneous (dark) clusters, "social" people as having many dots connecting them to other people outside their research group.



(a) (b)

Figure 5: Items: (a) highlight; and (b) pairwise comparison.

and columns, are highlighted (e.g., Figure 5.b shows the simultaneous highlight of George Robertson and Jeffrey Heer). The row of rectangle between the matrix and the labels then shows the correlation of the two current items similarities.

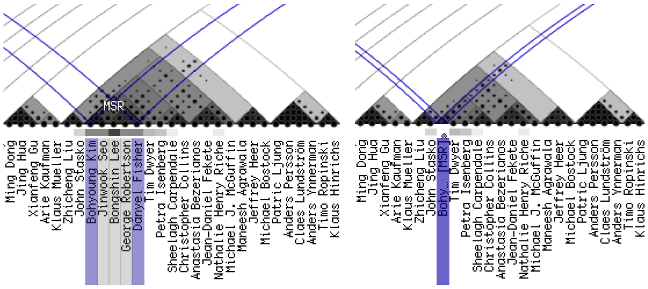
While hovering the matrix, the position of the cursor also designate a cluster: this cluster is linked to the pair of items highlighted because it is the most specific (smallest) cluster that contains both of them. We call this cluster the current cluster. The outline of this current cluster is highlighted, as well as the labels of the items it contains. It can then be manipulated as described below.

3.2.3 Cluster Aggregation

The clusters can be annotated: pressing the return key switches to the edition mode for the current cluster. The user can then give a label to this cluster. This label is displayed near the root of the cluster (Figure 6.a shows a cluster of 5 people labeled as “MSR”).

If the user clicks, the current cluster is folded, and its label (if it has been given any) is used to display the cluster among the items (Figure 6.b). The columns and rows of the similarity matrix are then replaced with an aggregation: it shows the similarity of the cluster with the other items using the linkage method of the AHC. That is an interesting property of AHC for us: clusters are, by definition, aggregations of their content, and the linkage method gives us the pertinent function to aggregate the information of items similarity when considering clusters. Once folded, clusters act exactly as items: it is possible to compare an item to a folded cluster, or two folded clusters together. It is also of course possible to fold a cluster that contains an already-folded sub-cluster.

If the user clicks on a folded cluster, it unfolds to reveal the detail of its content. Those transitions are continuously animated so that the user is not disoriented.



(a) (b)

Figure 6: Cluster (a) labeling; and (b) folding.

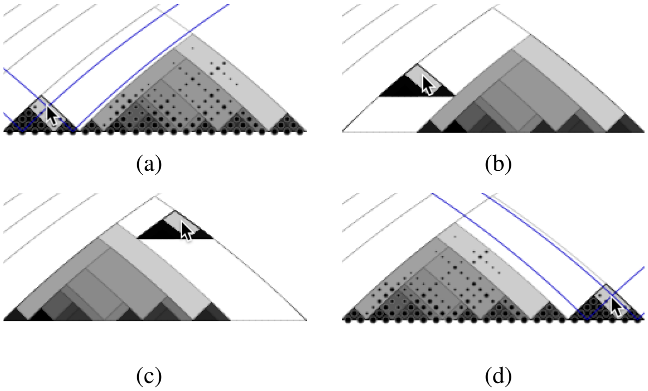


Figure 7: Cluster reordering using drag-and-drop: (a) initial state, (b–c) intermediate states, (d) final state with clusters swapped.

3.2.4 Tree Reordering

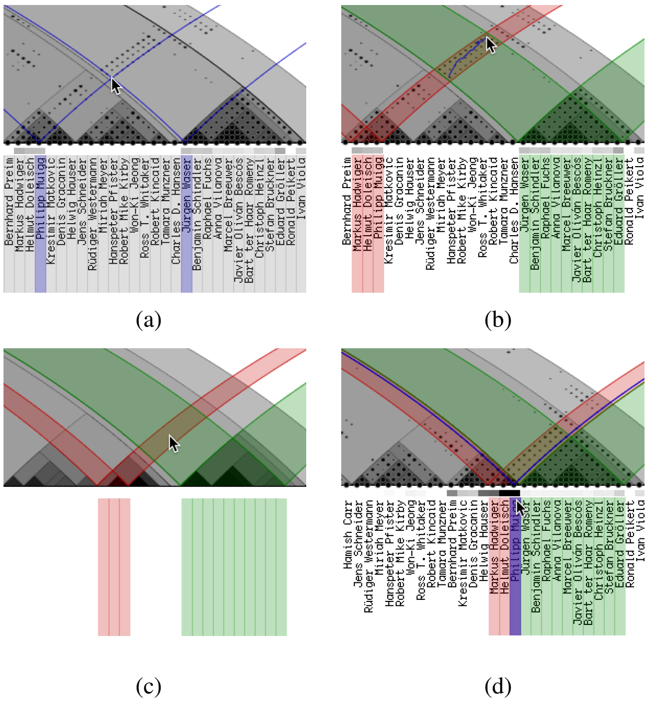
A drag-and-drop interaction allows to reorder the items by moving the current cluster (Figure 7). Since the order of items should be compatible with a permutation of the hierarchy of clusters coming from the AHC, the drag-and-drop is constrained. When dragging the current cluster towards the right (resp. left) there is no problem as long as the cluster being dragged is the left (resp. right) child of its parent: swapping it with its sibling effectively moves it to the right (resp. left). But if the current cluster is the right (resp. left) child of its parent, the system has to look for its smallest ancestor that is a left (resp. right) children of its parent to be able to swap it in the appropriate direction.

We provide a continuous control and feedback for an operation that is discrete in essence: the swapping of the two clusters. During the interaction (Figure 7.b and c), the position of the cluster being manipulated is constrained so that it stays within its parent and it does not overlap its sibling. Its movement has an inverted-V shape: it goes up towards the root of its parent; when it reaches this point, its brother pass below to the other side; and then it can go down to the other side. When the user ends the drag before reaching the new position, the movement is animated by the system: the cluster goes back to its starting point if it did not cross its sibling yet (Figure 7.b); or it goes on to the other side in the other case (Figure 7.c).

During the interaction and the animation, the dots giving the similarity and the labels are not draw because they rely on the order of the items, but this order is then ill-defined.

3.2.5 Clusters Bi-Manipulation

We provide an other way to reorder the tree: by specifying two clusters and then bringing them side by side. This interaction has been designed for a common use case: two similar clusters can be distant despite the optimal ordering. In this case, a rectangular pattern of dots can be seen away from the diagonal (Figure 8.a). The user can select the two corresponding clusters simultaneously using a crossing-based interaction (Figure 8.b): the smallest clusters including the projection of the cursor trace (in blue) on the rows and columns are selected (on the left, a cluster of three items is highlighted in red; on the right, a cluster of ten items is highlighted in green). The intersection of the two highlighted strips can then be dragged, and the projection of the cursor movements on the rows and columns control independently the two selected clusters (Figure 8.c). As during the simple drag, a continuous feedback is provided, and the similarity matrix and the labels are not displayed. The whole information is displayed anew upon the completion of the interaction (Figure 8.d).



(a)

(b)

(c)

(d)

Figure 8: Bringing two clusters side by side: (a–b) crossing-based bi-selection, and (c–d) bi-drag.

3.2.6 Partition Selection

An additional visualization is juxtaposed on the top left corner of the Dendrogramix (Figure 9). It plots the relationship between the number of clusters and their homogeneity. This visualization is interactive, and it makes it possible for the user to select a partition of the items. By dragging the labels, the user can choose a number of clusters on the horizontal axis, or an homogeneity on the vertical axis. The resulting partition is shown by a thick border and highlighted by a halo which separates the clusters more homogeneous than the cut (below the border) from the clusters less homogeneous (above the border).

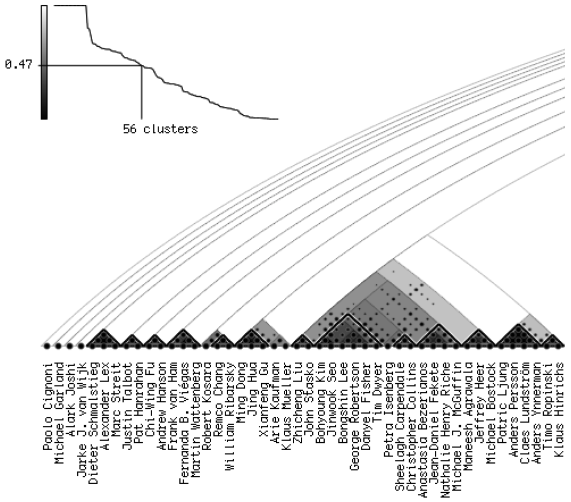


Figure 9: Selection of a partition.

3.3 Implementation

The Dendrogramix prototype[3] consists in less than 1500 lines of Python[4] code: about 100 for allowing reading various kind of data; 150 for performing the clustering (including the optimal order implementation); 550 for handling the graphical output; 450 for handling the interaction; and 250 lines of various utility functions.

The AHC is performed using the scipy.cluster.hierarchy[5] C extension for Python. The optimal order for the items is computed using the Bar-Jospeh et al. algorithm [2] that finds an optimum in the set all the permutations of the binary tree. Despite the dynamic programming techniques used, its complexity remains cubic (e.g., finding an optimal order for the 142 InfoVis authors takes 1.22 seconds on a recent computer, whereas finding an optimal order for a sub-set of 91 of them takes only 0.31 seconds). Some remarks though: first, those values could be cut by an order of magnitude by implementing the algorithm in C rather than Python. Since the complexity of the AHC is also polynomial (at best quadratic), the optimal ordering just faces the same bottleneck as the clustering itself. Second, an optimal ordering is not mandatory to build the Dendrogramix. As with the classical dendrogram, any order of the leaves compatible with the traversal of the cluster tree can be used. Using an optimal order has the property of bringing the densest parts of the similarity matrix as close as possible to its diagonal, thus concentrating the information towards the bottom of the Dendrogramix.

For the interactive graphic rendering, our prototype takes advantage of the hardware acceleration provided by the GPU by using the OpenGL[6] library. The interface with the windowing system is managed by the OpenGL Utility Toolkit (GLUT)[7]. The Python bindings to OpenGL and GLUT are provided by the PyOpenGL[8] package. Those choices make our prototype portable.

We use various techniques in order to perform the rendering sufficiently fast to provide smooth animations and fluid interactions while using an interpreted language. First, we use the programmable pipeline of OpenGL to perform various costly operations. The circular dots are rendered using simple squares that are textured procedurally as antialiased circles by a *fragment shader*. The non-linear transformations of the matrix are performed by a *vertex shader*. This shader compresses the matrix by altering the ordinate of each vertex while preserving its abscissa using the following transformation:

$$y' = \alpha \log\left(1 + \frac{y}{\alpha}\right) \qquad (1)$$

where y (resp. y') is the distance to the diagonal before (resp. after) the compression, and α a parameter that can be modified interactively by the user (the smaller is α, the more compressed is the matrix). The vertex shader also performs the folding of the matrix. The main program shares with the shader an array of coefficients that stores for each item how much its parent clusters have been folded. The shader uses this information to alter the geometry of the Dendrogramix so that the width of the column and the height of the row corresponding to an item are scaled accordingly (e.g., if a cluster of five items is folded, each item is scaled down by a factor of $1/5$, so that the folded cluster takes the same space as a single item).

The second technique we use is to cache various part of the rendering using OpenGL display lists. Those display lists are invalidated automatically by a state tracking mechanism. (e.g., the matrix

[3] Dendrogramix, <http://iihm.imag.fr/blanch/projects/dendrogramix/>.
[4] Python, <http://python.org/>.
[5] SciPy library, <http://scipy.org/>.
[6] OpenGL, <http://opengl.org/>.
[7] GLUT, <http://www.opengl.org/resources/libraries/glut/>.
[8] PyOpenGL, <http://pyopengl.sourceforge.net/>.

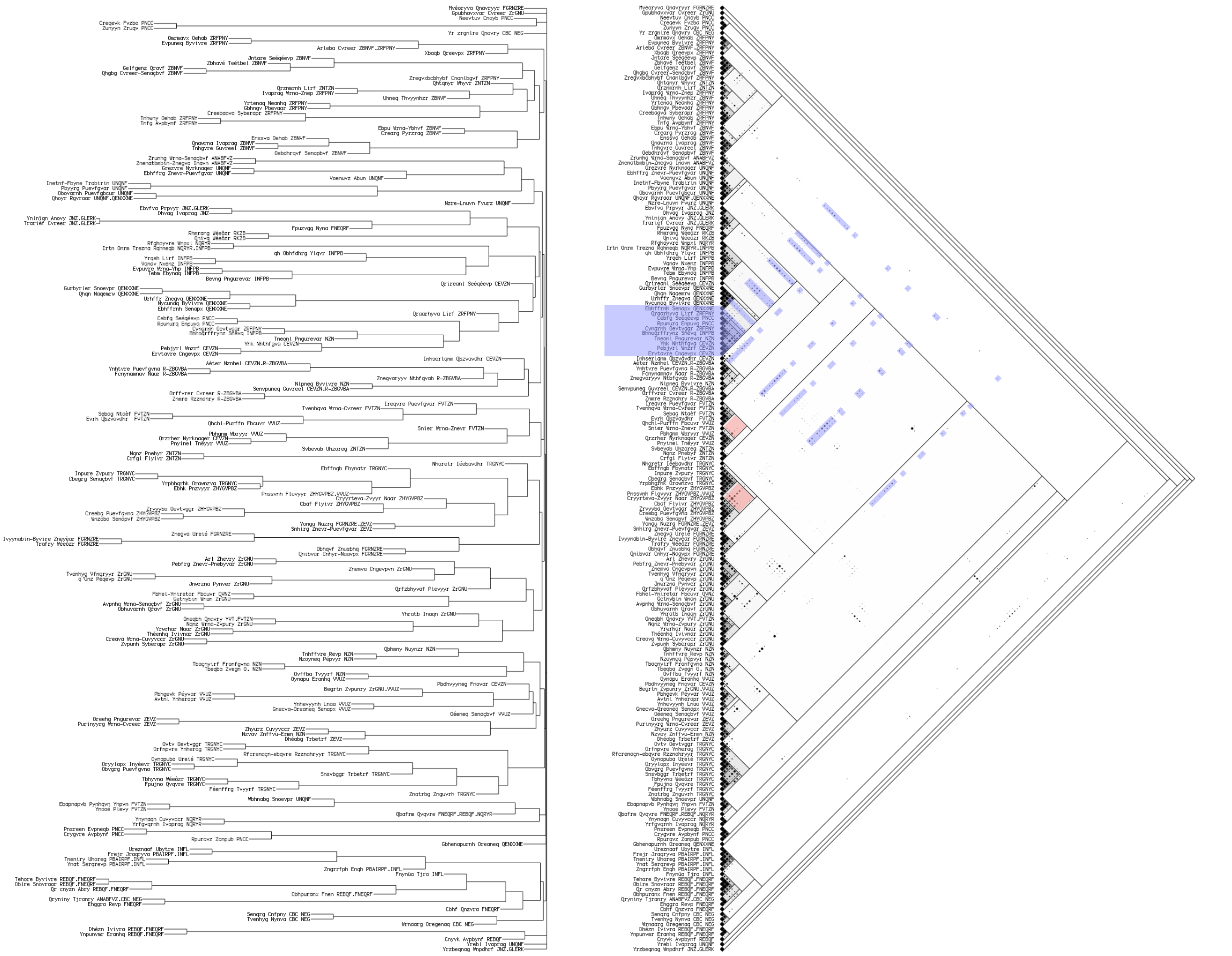


Figure 10: Collaborations between 189 researchers visualized with: (left) a classical dendrogram (CD); and (right) a Dendrogramix (DX) with insights revealed by patterns highlighted in blue (top, "grid" pattern) and red (middle, "cross" patterns). The names of the researchers have been obfuscated and the figures rotated to better fit on the paper.

is rendered in direct mode once at the beginning of the program and that rendering is reused as long as the leaves are not reordered). Using those techniques, we are able to render a Dendrogramix at more than 30 frames per second when necessary.

Finally, our prototype is also able to generate a static vector graphic output (in SVG format) suitable for integration with various graphical tools. Figure 1 is an example of such a vector graphics image, whereas the other Figures use screen captures of the raster graphics produced by OpenGL in order to show the feedback of the interaction techniques.

4 Evaluation

To evaluate the Dendrogramix visualization, we have conducted an experiment that compares it to the classical dendrogram. The experiment is a case study in which experts where shown a dataset related to their area of expertise and asked to think-aloud during their exploration of the dataset. We then used an insight-based evaluation method [14, 12] to quantify the performance of the two visualization techniques. We used this methodology because we knew that some specific tasks would not be possible at all with the classical visualization, and thus including those tasks in a controlled experiment would have made no sense, but excluding them would not have demonstrated the whole potential of Dendrogramix.

4.1 Data set

In this case study, users are a set of 6 senior researchers of a computer science research center that employ about 200 permanent researchers. They are either deputy director of the research center, or leading a research team of 6 to 12 permanent researchers (as team leaders, they are members of the scientific council of the research center). Those positions give to all of them a very good knowledge of the structure of the research center and of the research themes of the various research teams.

The data submitted to their assessment is a clustering of the permanent researchers of the research center (Figure 10). Those 189 researchers are each characterized by how many publications they have coauthored with each possible coauthor involved in a publication of the research center, i.e. a vector of length 2688 (the number of unique authors found in the research center bibliographical database). The distance used to compute the similarity between authors is the cosine distance. The generalization of this similarity to groups uses the average method.

The list of authors was deduplicated manually (112 duplicates where found amongst the 2800 names present in the database). The publication database included 3245 references for a four years period (2010–2013).

4.2 Visualizations

The resulting clustering was presented graphically using two techniques: the classical dendrogram (CD) and the Dendrogramix (DX). They both fitted on a 1920 × 1200, 23" Apple Cinema HD Display LCD screen. Figure 10 shows on the left the classical dendrogram and on the right the Dendrogramix visualization used (both rotated to better fit on the paper). The labels of the items are the names of the researchers followed by the names of their teams in capital letters. Those strings are obfuscated on Figure 10 for anonymization purpose, but they were displayed in clear for the case study.

The interaction techniques proposed with the visualizations were restricted to the bare minimum: the items were highlighted on hovering, and a search field allowed to enter a string in order to highlight all its occurrences in the items labels. That allowed the users to quickly find specific researchers or research teams. This lack of interaction is also deliberate to put the two techniques on par: classical dendrograms do not come with interactions and we did not want to favor our technique.

4.3 Protocol

The experiment consisted in three phases for each participant: first an introduction to the data set and the visualization techniques was made; then the participants were invited to explore the data and report their findings by thinking-aloud; and finally a quick wrap-up was conducted. The whole session lasted one hour.

The first phase lasted about 10 minutes during which the source of the data and the construction of the HCA was elicited. As they were all computer scientists, all participants were familiar to some extent with HCA and classical dendrograms.

In the second phase they used each of the visualization in turn during 20 minutes. They were instructed to first look at themselves; then at their team; then, at their will, at other teams or any other subject of interest to them. Half of the participants used CD first, then DX, whereas the other half used the two techniques in the other order. They reported their findings vocally and were encouraged to ask questions and formulate hypothesis. We took notes and recorded the audio of those interviews for further analysis and coding.

Finally, we asked them for their observations and comments about the two visualizations in a 10 minutes wrap-up session at the end of the interview.

4.4 Results

To analyze the results of this experiment, we used the insight-based methodology described by North, Saraiya et al. [12]: we first counted the various insights found by the participants, then characterized them. The two main characteristics used are the category and the domain value, with definitions adapted from Saraiya, North et al. [14] (emphasized text below). We used 4 categories: *overview (overall distributions of* [researchers]*), patterns (identification or comparison across data attributes), groups (identification or comparison of groups of* [researchers]*), and details (focused information about specific* [researchers]*).* The domain value is rated between 1 and 5: *trivial observations earn 1-2 points, insights about a particular* [structure] *earn an intermediate value of 3, and insights that confirm, deny, or create a hypothesis earn 4 or 5 points.* The ratings have been established by asking the participants during the wrap-up session for the relative importances of their findings. All the participants gave an order mostly consistent with the other ones. The time spent was not considered as a characteristic since it was specified a priori (20 minutes per technique).

	CD		DX		common
category	count	value	count	value	count
overview	6	6	7	7	5
patterns	3	8	14	45	3
groups	14	34	17	43	14
details	13	19	27	67	13
total	36	67	65	162	35
per insight		1.86		2.49	
per minute	0.30	0.56	0.54	1.35	

Table 1: Number of insights and their value per visualization technique, dendrogram (CD) and Dendrogramix (DX); and category.

Table 1 gives an overview of the insights' counts and values broken-up by techniques (columns) and categories (lines). The last column reports the count of insights that are common to both techniques. This gives the very first observation about the results: almost all the insights found with CD are also seen with DX. Since half of the participants were using CD first then DX, it means that for them, all the insights they found with CD were also observable with DX, and they found new ones with DX. On the other hand, participants that started with DX did not found any new insight when switching latter to CD.

The second observation is about the values of insights: DX gave insights scoring 2.49 on average vs. 1.86 for CD. So DX not only gives more insights (1.8 times more) than CD, but those insights are of higher value. This difference in quality can be illustrated with examples of insights. Participants often started their investigations by looking-up their own name, then looking at their team, then looking for other structures. Thus, they generally started by insights on items, and then moved towards higher levels of details.

4.4.1 Details

Insights about details mainly occurred to participants looking up their own names. Most of those insights came while looking for the closest collaborators suggested by the clustering. Those insights are of low value: they are not surprising; but they gave participants trust in the visualizations. Other insights came after having investigated more high level structures and then coming back to individuals. They were less trivial, e.g., "the leader of this research group does not collaborate with any of the group researcher, is this deliberate?" This second type of insights was made only with DX.

4.4.2 Groups

Insights about groups were mainly observations about the structure of research teams. Using the search field, participants highlighted the members of teams they knew well (their own team and teams familiar to them). Various kind of structures are observed with both visualizations: a coherent team forming a unique cluster; teams clustered into 2 or more main groups; and teams with members disseminated in various other groups. People were sometime surprised by those findings, but found explanations to those structures (often linked to the history of the team).

4.4.3 Patterns

The patterns found are interesting: they are all insights about the relationship between researchers and groups of researchers. Such insights are found almost exclusively with DX.

A simple pattern is a group of 10 people from various teams clustered together mostly without any research interest in common. All participants noticed this group and found an explanation for it: it is the board of deputy directors of the research center. The co-publications linking them are the activity reports of the center that are included in the bibliographical database used for the visualizations. Moreover, many participants noticed that this group is also linked to researchers outside the group in a pattern that forms a "grid" as highlighted in blue on the Dendrogramix of Figure 10.

Those researchers are in fact team leaders (and, as such, co-authors of the activity reports), but are more tightly linked to their teams than the deputy directors since they still have a research activity.

Another frequent insight found is related to a "cross" pattern as highlighted in red on the Dendrogramix of Figure 10. That cross links two otherwise disjoint clusters. One occurrence of the pattern can be explained by the presence of two research engineers, each in one team, who link those teams because they work together on a research infrastructure used by both teams. The other occurrence has a similar explanation.

Other insights came after looking at the largest dots (or groups of dots) found far from the base of DX. They represent people that links two groups of researcher that are not otherwise connected (sometimes the two groups are sub-parts of a single research teams).

4.4.4 Overview

Few overview insights were given: they related mostly to the size of the research center, the number of teams and their relative sizes.

4.5 Discussion

As already stated, almost all insights found with CD were also found with DX. The two last lines of Table 1 show the superiority of DX over CD: DX gives more insights per unit of time (.54 vs. .30 insight per minute) but also provides better insights (2.49 vs. 1.86 points) resulting in a 2.41 higher "domain value throughput" (1.35 vs. .56 points per minute).

While overall superior, the Dendrogramix technique is especially good at showings patterns linking items to clusters they do not belong to. The classical dendrogram lacks this expressiveness: once an item is in a cluster, its links to other clusters are not considered.

However some limitations of this work should be acknowledged: the first one is that this technique is well suited only for a strict hierarchical clustering. The Dendrogramix relies on the tree structure for the visualization as well as for the interaction. It may not be possible to adapt this technique to other clustering methods in which clusters can overlap.

Finally, the methodology used for the evaluation could be disputed. We chose the insight-based methodology because the two techniques are not directly comparable: some tasks (e.g., items comparisons) can just not be performed with CD. We could have opted for a more controlled experiment, but to do so, we would have had to enhance the CD. One solution would have been to juxtapose the similarity matrix to the dendrogram, but this is not a viable option in a realistic scenario: the space next to the dendrogram is often reserved for a heat map displaying the raw items. The insight-based methodology allowed us to compare the techniques as they are.

5 Conclusion and Future Work

We have introduced Dendrogramix, an interactive visualization that allows the presentation and the exploration of the result of an AHC. This visualization gives clues to the user about the way clusters have been generated and about the role of the particular items in this construction. The interaction techniques we have presented makes the exploration and the annotation of the tree of clusters possible, thus helping users make sense of them.

We plan to extend this work in two main directions. First, we would like to use Dendrogramix to provide a tool that would allow the comparison of different clustering algorithms. Indeed, the impact of the similarity metric and the linkage method used on the result of an AHC is difficult to grasp, but superimposing the input (the similarity matrix) and the output (the hierarchy of clusters), as the Dendrogramix does, helps to understand the clustering process. Displaying multiple Dendrogramix of the same data set clustered with different methods may help users gain a finer understanding of the various clustering techniques.

Second, we want to explore the use of Dendrogramix on the result of co-clustering algorithms. A possible way to do that would be to juxtapose 2 Dendrogramices with a heat map as it is often done for classical dendrograms: one on the top that would allow to reorder and fold the columns of the heat map, and one on the side that would allow to manipulate its rows. Those future works will be conducted in cooperation with machine learning specialists and experts from a domain using (co-)clustering algorithms heavily (namely proteomics) in order to evaluate formally the advantages of such extended visualizations. Initial feedbacks are very promising.

Acknowledgments

Thanks to Michael McGuffin, Nathalie Henry-Riche, Jean-Daniel Fekete for early feedback. Many thanks to the various people cited in the examples without having been asked.

References

[1] G. Apitz and F. Guimbretière. CrossY: a crossing-based drawing application. In *Proc. ACM UIST 2004*, pages 3–12, 2004.

[2] Z. Bar-Joseph, E. D. Demaine, D. K. Gifford, A. M. Hamel, T. S. Jaakkola, and N. Srebro. K-ary clustering with optimal leaf ordering for gene expression data. *Bioinformatics*, 19(9):1070–8, 2003.

[3] G. Bisson and R. Blanch. Improving visualization of large hierarchical clustering. In *Proceedings of the 16th International Conference on Information Visualisation (IV 2012)*, pages 220–228, 2012.

[4] R. Blanch and É. Lecolinet. Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. *IEEE Trans. on Vis. and Comp. Graph. (Proc. InfoVis 2007)*, 13(6):1248–1253, 2007.

[5] J. Chen, A. M. MacEachren, and D. J. Peuquet. Constructing overview + detail dendrogram-matrix views. *IEEE Trans. on Vis. and Comp. Graph. (Proc. InfoVis 2009)*, 15(6):889–896, Nov. 2009.

[6] N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Trans. Vis. Comput. Graph.*, 16(3):439–454, 2010.

[7] J. Gower and P. Digby. Expressing complex relationships in two dimensions. In V. Barnett, editor, *Interpreting Multivariate Data*, pages 83–118. John Wiley & Sons, Inc., 1981.

[8] N. Henry and J.-D. Fekete. MatLink: Enhanced matrix visualization for analyzing social networks. In *Proc. Interact'07*, pages 288–302, 2007.

[9] N. Henry, J.-D. Fekete, and M. J. McGuffin. NodeTrix: Hybrid representation for analyzing social networks. *IEEE Trans. on Vis. and Comp. Graph. (Proc. InfoVis 2007)*, 13(6):1302–9, 2007.

[10] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *Proc. IEEE PacificVis 2012*, pages 1–8, 2012.

[11] M. J. McGuffin and J.-M. Robert. Quantifying the space-efficiency of 2D graphical representations of trees. *Information Visualization (IVS)*, 9(2):115–140, 2010.

[12] C. North, P. Saraiya, and K. Duca. A comparison of benchmark task and insight evaluation methods for information visualization. *Information Visualization*, 10(3):162–181, 2011.

[13] S. Rufiange, M. J. McGuffin, and C. P. Fuhrman. TreeMatrix: A hybrid visualization of compound graphs. *Computer Graphics Forum (CGF)*, 31(1):89–101, 2012.

[14] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Trans. on Vis. Comp. Graph.*, 11(4):443–456, 2005.

[15] B. Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8):57–69, Aug. 1983.

[16] P. H. A. Sneath. The application of computers to taxonomy. *Journal of General Microbiology*, 17(1):201–226, 1957.

[17] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

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

[19] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *Proc. IEEE InfoVis 2005*, pages 57–64, October 2005.