Stacked Trees: a New Hybrid Visualization Method

Gilles Bisson
CNRS - AMA team
LIG laboratory - UMR 5217
UFR IM2AG - BP 53 - F-38041 Grenoble Cedex 9
+33 (0)456 52 03 07
gilles.bisson@imag.fr

Renaud Blanch
UJF-Grenoble 1 - IIHM team
LIG laboratory - UMR 5217
UFR IM2AG - BP 53 - F-38041 Grenoble Cedex 9
+33 (0)476 51 43 65
renaud.blanch@imag.fr

ABSTRACT

In this paper, we introduce a new Focus+Context visualization technique, named “Stacked Trees”, allowing us to explore large dendrograms produced by hierarchical clustering. This approach displays up to fifty thousands nodes on a standard-sized screen.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI), I.2.6 [**Learning**]: Induction – *Hierarchical Clustering*.

General Terms: Algorithms

Keywords

Hierarchical Clustering, Stacked Trees, Large Dendrograms.

1. INTRODUCTION

Clustering is a classical explanatory approach [1], helping to explore information contained in large databases, by organizing the data into clusters based on a similarity measure. A practical drawback of this approach lies in the fact that validating the clusters is not straightforward. Although several automatic methods exist [4] to evaluate the relevance of the clustering results, when dealing with a new problem, the most efficient solution is to work with an expert of the target domain who analyzes manually the clusters in order to interpret them and to identify the meaningful information. Thus, to be successful, one needs to provide to this expert some visualization tools.

In such context, Agglomerative Hierarchical Clustering (AHC) is a well-suited method to provide to the users some relevant information for analyzing data. Indeed, the AHC procedure organizes data in an intuitive and interpretable way for human being, namely a binary tree or *dendrogram*. In such structure all degree of generality are present from the basic instances to the most general cluster. However, visualization of large sized trees is known to be difficult since the number of leaves grows exponentially with the depth of the tree. Practically, when dealing with a dendrograms containing more than a few hundred leaves, any *node-link representation* of the tree becomes unfeasible. Of course, it is always possible to display at one time only a subpart of the structure and to explore the other parts, through a combination techniques such as: filtering, distortion, zooming or panning [10]. However, as emphasized by [7] these approaches involve some design tradeoffs and constraints for the user: for instance, the need of integrating the different subviews in the zooming approaches or the need of understanding a distorted view. Therefore, providing a really *static view* able to present a large amount of data in a non-ambiguous, uncluttered, scalable and aesthetic way is a worthwhile challenge.

Before outlining our visualization and comparing it to the existing ones, we first explain rapidly a chemoinformatics application whose analysis led us to the current proposal. The *High-throughput Screening* process is designed to quickly test, by using robotic devices, the bioactivity of a set of molecules, organized within a *chemical library*. Each test highlights some tens or hundreds of *active molecules* (named *hits*) representing generally a very small percentage of the chemical library (<<1%). In this context, it is crucial to provide to the chemists some interactive tools enabling them to pinpoint the location of the active molecules within this chemical space and to ease the search for related molecules in order to help the synthesis of more efficient compounds. Here are some of the typical queries to address:

- To identify the position of the hits with respect to the main clusters to see if their chemical structures are related or not.
- To display the shared chemical/physical properties (mass, logP, etc.) of the hits and more generally to see the properties associated to a given cluster.
- To detect the unexpected clusters and to analyze if they bring some new information or they just reveal errors or incompleteness in the description of instances.

Obviously, these tasks are not specific to chemistry: exchanging the terms “hits”, “molecules”, … by the equivalent objects of another application domain, leads to the same kinds of tasks. In every case, the goal is to understand what are the *environment* and the *properties* of a set of objects (instances/clusters) with respect to the others. In terms of visualization techniques, this means that the user must be able to access simultaneously the *local information* contained in the leaves of the hierarchy, (here, the molecules features), and also the *global information* characterized by the medium and higher levels of the hierarchy in order to identify the relationships between the clusters generated by the algorithm. This could be seen as a classical *Focus+Context* [7] problem, but in our task we do not have a single focus but several (the hits) at the same time and all the molecules occurring in the same cluster are, *a priori*, equally interesting to explore.

The rest of the paper is organized as follows. In Section 2, we provide a short state of the art concerning the current techniques to visualize a dendrogram. Then, in Section 3 we will present a new visualization method named “Stacked Trees” and we will discuss its benefit in terms of *information density*. Finally, in Section 4 we will present the prototype we developed and we will describe the interactions between the user and this tool.

AVI ‘12, May 21-25, 2012, Capri Island, Italy

2. STATE OF THE ART

Hierarchies are general and intuitive structures allowing us to represent a wide range of phenomena. Thus, countless studies have been conducted in InfoVis to visualize large hierarchies. The recent survey of [10] provides a wide overview of the representation paradigms. One can divide most of the techniques into three categories: *node-links*, *space filling* and *hybrid*.

Hyperbolic trees [9] can be seen as a specific implementation of the *fisheye framework* [7]. They belong to the *node-links* techniques and consist in projecting the whole hierarchy in a non-Euclidean space. However, even if these approaches can be used to display a larger number of objects than a classical "Euclidian" hierarchy, the filling of the available space remains rather small and limits the maximum number of objects; in the best cases up to several thousands as in FSVIZ [5]. More recently, *SpaceTrees*, introduced by [11], proposes a dynamic approach to the visualization problem in which the different parts of the hierarchy are dynamically reconfigured while browsing through the folding/unfolding of its subtrees. However, if these approaches allow keeping an explicit representation of the relationships between the clusters, 1) we need to browse the structure to access the information contained in the leaves and 2) the screen layout limits the number of visible items to some thousands.

TreeMaps [12] belong to the *space filling* category. In this layout, the hierarchy levels are represented by a sequence of nested rectangles allowing both an optimal use of the display space and the visualization of the low-level information (instances) in a homogeneous way. Moreover, by using the power of GPUs, we can manage interactive and zoomable maps containing millions of objects as in [3], [6]. Nevertheless, the TreeMaps have two drawbacks: on the one hand, for the novice user, the hierarchical structure is hard to perceive since it is very difficult to distinguish among the different levels of the dendrogram; and on the other hand, the relative position of the blocks (i.e. clusters) on the screen is not very intuitive and requires some practices.

Elastic Hierarchies [13] are a *hybrid* approach combining a node-link representation and a set of TreeMaps. In this tool, as with the SpaceTrees, the user interactively adapts the layout in order to fulfill her/his needs for each subpart of the hierarchy. In this way, one theoretically preserves the advantages of both approaches: interpretability of the hierarchical structure and compactness of the data. However, the criticism about the arbitrary position of the clusters in the TreeMaps remains and the optimal use of this kind of tool requires some training from the user.

3. STACKED TREES

The *Stacked Trees* method proposed in this paper also belongs to the hybrid representation family, but it is based on a simpler paradigm than the Elastic Hierarchies in the sense: 1) that the user does not have the full ability to select his/her representation layout at each level of the dendrogram 2) that the visualization layout remains close to the one of classical dendrogram, thus simplifying the learning curve and 3) that the representation we use to compact the information is not based on a 2D structure as in the TreeMaps but rather on a simplified *1D structure* that we call "stacks". It worth noting that similar ideas were proposed by [8] in order to visualize in a compact way the content of large multi-attributes database. However, this previous work was not based on notions of hierarchies/dendrograms nor clustering.

The basic idea of the Stacked Trees is the following: as we discussed in the introduction, when a chemist wants to analyze a hierarchical clustering she/he needs mainly to access simultaneously 1) to the local information contained in the leaves of the dendrogram (here the features of the molecules) and 2) to highest levels of the dendrogram in order to grasp the overall organization of the clusters. In other terms, the medium part of the dendrogram is not very informative. Thus, as shown in Figure 1, the idea is to suppress this part and to organize all the leaves (instances) belonging to a given *subtree* in the form of a vertical 1D structure corresponding to a *stack*. Therefore, in our terminology the words *stack*, *subtree* and *clusters* are synonyms.

Beyond the compactness argument, this stack-based organization has also a very interesting property: it allows the expert to dynamically reorder the instances on the screen according to her/his needs. Indeed, in various domains, *features* describing the instances are numerical or, more generally, they belong to an "ordered type". This is the case in our application in which molecules can be ranked along many dimensions such as: bioactivity intensity, mass, hydrophobicity, LogP, etc. Thus, 1D representation is perfectly suited to organize the information in a comprehensive way for the user; for example, to visualize how the set of molecular masses are distributed within a cluster or to show the similarity between hits and other molecules. This representation is also coherent to display clustering results since they correspond to a (partially) ordered sequence of instances. In this way we avoid the problem we evoked about the TreeMaps concerning the arbitrary — or at least difficult to explain— positions of the clusters/instances in the maps.

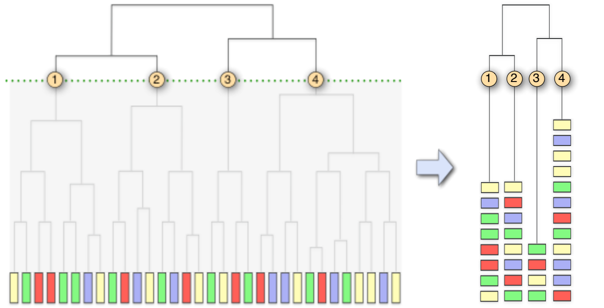


Figure 1. Stacked Trees just keep the highest levels of the dendrogram and leaves are organized into vertical "stacks".

Evaluating visualization method usefulness is a rather difficult task and many metrics has been proposed [2], [10]. In this paper, according to our goal of maximizing the quantity of information displayed at the same time, we compare the number of instances and the number of levels that can be simultaneously displayed in the case of a "well balanced" hierarchy (Table 1). Our baseline is a 24-inch classical screen containing about 2Mpxs.

First of all, we need to decide the minimum number of pixels needed to convey the information contained in an instance. To represent in a compact way a tuple <*feature, value*> coding one information of a given instance, we can use a *color code*, that can be either discrete or a gradient. In practice, to be easily visible (and selectable with a standard pointer: mouse, stylus or finger), we make the assumption that each value must fill at least an area of about 3x3 pixels. Moreover, one needs to separate the instances with 1 pixel in order to stay readable. Consequently, we will devote an area of 16 pixels (4x4) to display each instance.

As we see in Table 1, in the TreeMaps paradigm, the whole screen can be used to display the instances since the hierarchical structure is "implicit". This allows representing up to 125.000 instances at the same time. For the node-links representation we consider that 1) we use a simple Euclidian projection, 2) that there is not overlapping between links and nodes and 3) that the nodes are aligned along the 4 edges of the screen corresponding to a total width of 5000 pixels. Under these hypotheses, up to 1250 instances (items) can be displayed at the same time.

Table 1. Comparison of the information density.

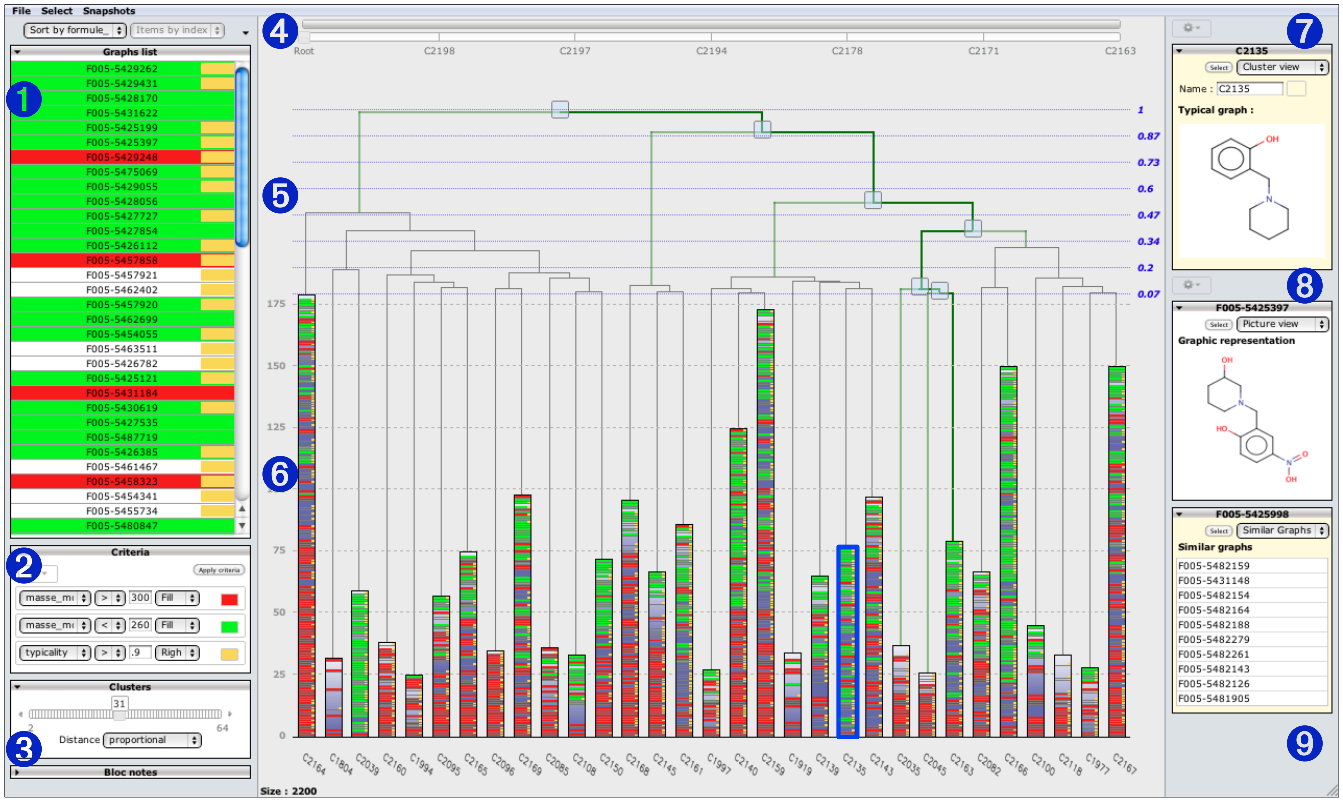
Criteria	TreeMaps	Nodes-links	Stacked Trees
Usable area for data (in blue)	Instances area	Instances area Tree structure	Tree structure Instances area
#pixels	Fullscreen: 2Mpx	Edges: ~5kpx	Bottom: ~0.8Mpx
#items	125.000	1250	50.000
#levels	~17	~10	~9

Finally, with the Stacked Trees representation, the calculus is little bit more complex. On the one hand, as in our hypothesis each stack has 4 pixels width (the size of an instance), it is not possible to put more than 500 stacks on a 24" screen of 2000 pixels width. On the other hand, the vertical area dedicated to display the stacks represents about 80% of the screen, as the top of the screen is needed to draw the structure of the highest levels of the dendrogram. Furthermore, even with a well-balanced hierarchy, the stacks will fill, on average, only half of this area since the clusters cannot be equally sized (see Figure 2 which is a typical example). By consequence, the display area devoted to the instances is about 40% of the screen that is corresponding to 400 pixels height and up to 100 instances per stack. So, about 50.000 instances can be simultaneously shown on the screen.

4. DESCRIPTION OF THE PROTOTYPE

Figure 3 is a screenshot of the prototype we implemented for our chemical application. The central area is composed of two parts corresponding to the Stacked Trees representation. At the top (part number 5) one finds a classical hierarchy going with a standard *combination similarity* scale on the right allowing us to measure the distance between clusters. At the bottom (part 6) is a collection of stacks. The number of stacks to show is dynamically controlled by the user through a slider (part 3) allowing an interactive adjustment of the *cut level* of the hierarchy. In this screenshot, this value can be adjusted between 2 and 64 but the upper limit depends on the width of the screen. The height of each stack is proportional to the number of instances in the subtree. The molecule names of the currently selected stack (here, C2135) are displayed in a scroll list (part 1) using a classical *Focus+Context* method [7]. Thus, to get the name of a molecule in a stack, the user has to select the corresponding rectangle and the molecule will be highlighted in part 1, the selection process being a bidirectional one. The stacks and molecule currently selected are corresponding to the *focus* in the interface, the other stacks and their relational structure being the *context*.

Visualization of the instances: in part 2, the user can declare some *display rules*, in the form of triplets <*feature, selector, value*>, then she/he can associate to each of these rules a *position* and a *color* changing the way instances are displayed in parts 1 and 6. For instance, in Figure 3, we colored molecules whose *typicality* (i.e. their similarity with the cluster prototype) is smaller than 0.6 (orange=right) and whose mass is smaller than 260 (body=green) or greater than 300 (body=red). Thanks to this mechanism, it is possible to represent a wealth of information in a very compact


File Select Snapshots

Sort by formule_ | Items by index

Graphs list

F005-5429262
F005-5429431
F005-5428170
F005-5431622
F005-5425199
F005-5425397
F005-5429248
F005-5475069
F005-5429055
F005-5428056
F005-5427727
F005-5427854
F005-5426112
F005-5457858
F005-5457921
F005-5462402
F005-5457920
F005-5462699
F005-5454055
F005-5463511
F005-5426782
F005-5425121
F005-5431184
F005-5430619
F005-5427535
F005-5487719
F005-5426385
F005-5461467
F005-5458323
F005-5454341
F005-5455734
F005-5480847

Criteria

Apply criteria

masse_m	>	300	Fill	🟥
masse_m	<	260	Fill	🟩
typicality	>	.9	Righ	🟨

Clusters

31

2 64

Distance: proportional

Bloc notes

C2135

Select | Cluster view

Name : C2135

Typical graph :

F005-5425397

Select | Picture view

Graphic representation

F005-5425998

Select | Similar Graphs

Similar graphs

F005-5482159
F005-5431148
F005-5482154
F005-5482164
F005-5482188
F005-5482279
F005-5482261
F005-5482143
F005-5482126
F005-5481905

Size : 2200

Figure 2.Here is a screenshot of our current prototype. The blue numbers highlight the main parts of this interface whose roles are explained in the paper. In this figure, 2200 molecule are simultaneously displayed with for each one up to three properties.

way and to provide a visual feedback to queries such as: "*Where are located the hits with respect to the molecule having a given mass and hydrophobicity?*", "*What is the homogeneity of the chemical space in terms of masses*", etc. In chemistry it is also crucial to provide the experts with an access to the 2D structure of the molecules. Here, the user can visualize (part 7) the most typical molecule of the currently selected stack and she/he can create several display editors (parts 8) to compare molecules. The drawing of the molecules is performed "on the fly" by the application server using the tool MARVIN by Chemaxon. Some other possibilities are also offered, such as the ability to view a list of the "nearest neighbors" of the selected molecule (9).

Instances ranking in the stacks: to help to understand the tree structure and the meaning of the clusters, it is important to allow the user to change her/his point of view on the data. Thus, in our prototype, the expert can select a (ordered) feature and reorganize its instances by ascending (or descending) values within the stacks, through the menu at the top of the part 1. In this way, she/he can display, for example, the mass of the molecules to see if "hits" are corresponding to low or heavy molecules.

Navigation within the hierarchy: even if the main objective of Stacked Trees is to provide a *static view* able to display a large amount of instances, it is nevertheless useful to allow the user to explore a specific subpart of the clustering, to access to more detailed information. Here, the user can, at any time, select a new root for the hierarchy by simply clicking on one node in the part 5 of the interface. It is thus possible to recursively go down into the dendrogram to explore the subclasses, the two bars (part 4) indicating the position, width and depth of the area currently explored. Of course, when going down into the hierarchy, stacks contain fewer and fewer molecules and ultimately, when the number of molecules equals to the number of visible stacks (part 3), the behavior of the interface becomes similar to the one of a standard tree viewer: each stack is corresponding to a single leaf (Figure 5). In this sense, Stacked Trees approach can be seen as a natural generalization of the standard node-link representation.

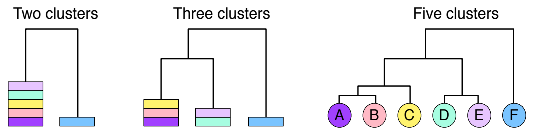


Figure 3. Turning a Stacked Tree into a classical dendrogram.

The current prototype has been implemented as a Javascript Web application, which can be used through any browser compliant with the W3C standards. The layout and graphical properties of all the elements of the interface are controlled through CSS. The communication between the interface and the data files containing the *domain specific* information (namely: hierarchical structure, similarity matrix, <feature, value> tuples coding the instances and the external visualization tools) is managed by PHP.

5. CONCLUSION

In this paper, we present a new hybrid visualization technique, named *Stacked Trees*. This visualization paradigm has several nice advantages. Firstly, its layout is simple to understand and it can be seen as a natural generalization of the classical (node-link) dendrogram visualization, thus easing the learning curve for the user. Secondly, the information density is quite large allowing us to deal with up to 50 000 instances and thus providing a good alternative to TreeMaps. Thirdly, from the complexity point of view, all the procedures used to draw the nodes on screen are linear in terms of number of instances. Finally, although this work has been initially done to help the analysis of clustering results in chemistry, the approach is generic and modular enough to bring a new solution to many other application domains.

6. REFERENCES

[1] Berkhin P. 2006. Survey of clustering data mining techniques. *In Grouping Multidimensional Data*. Springer Berlin Heidelberg, 25-71.

[2] Bertini E. 2011. Quality Metrics in High-Dimensional Data Visualisation: An Overview and Systematization. *IEEE Transactions on Visualization and Computer Graphics*. Vol 17, No 12, 2203-2212.

[3] Blanch R. and Lecolinet E. 2007. Browsing Zoomable Treemaps: Structure-Aware Multi-Scale Navigation Techniques 2007. *IEEE Transactions on Visualization and Computer Graphics 13(6), Proceedings of IEEE InfoVis 2007*, 1248-1253.

[4] Candillier L., Tellier I., Torre F. and Bousquet O. 2006. Cascade Evaluation of Clustering Algorithms. *In Proceedings of the 17th European Conference on Machine Learning ECML'2006*, Berlin, Germany, 18-22 september 2006, LNAI 4212, 574-581.

[5] Carriere J. and Kazman R. 1995. Interacting with Huge Hierarchies: Beyond Cone Trees. *In Proceedings IEEE Information Visualization*, 74-81.

[6] Fekete J-D. and Plaisant C. 2002. Interactive Information Visualization of a Million Items. *In Proceedings of the IEEE Symposium on information Visualization (InfoVis).* Washington, DC, 117-126.

[7] Furnas G. W. 2006. A fisheye follow-up: further reflections on focus + context. *In proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06),* Rebecca Grinter, Thomas Rodden, Paul Aoki, Ed Cutrell, Robin Jeffries, and Gary Olson (Eds.). ACM, New York, 999-1008.

[8] Keim D., Hao M., Dayal U., M.Hsu, J. Ladisch 2001. Pixel Bar Charts: A New Technique for Visualizing Large Multi-Attribute Data Sets without Aggregation. *In Proceedings of the IEEE Symposium on Information Visualization 2001 (InfoVis)*. IEEE Computer Society Washington DC, USA.

[9] Lamping J, Rao R. and Pirolli P. 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *In Proceedings of ACM Conference Human Factors in Computing Systems*, 401-408.

[10] Landesberger (von) T., Kuijper, A., Schreck T., Kohlhammer J., van Wijk, JJ. Fekete, J-D and Fellner DW. 2011. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum*. Volume 30, Issue 6, 1719-1749.

[11] Plaisant C., Grosjean J. and Bederson B. 2002. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. *IEEE Symposium on Information Visualization (InfoVis)*. Washington DC, 57-66.

[12] Shneiderman, B. 1992. Tree visualization with tree-maps: 2-D space-filling approach. *ACM Transactions on Graphics,* 11(1), 92-99.

[13] Zhao S., McGuffin, M.J., Chignell, M.H. 2005. Elastic hierarchies: combining treemaps and node-link diagrams. *In Proceedings of the IEEE Symposium on Information Visualization 2005 (InfoVis)*, 57-64.