

Bubble Hierarchies and Pythagoras Trees: Two Perspectives on Hierarchical Data

Michael Burch

Fig. 1. Generalized Pythagoras trees and bubble hierarchies for visually depicting hierarchical data: Although the hierarchy visualizations look differently, they can support a data analyst by providing different views on the same dataset.

Abstract—In this paper we describe an interactive visualization for displaying hierarchical data in several perspectives. To reach this goal we combine bubble hierarchies as well as the generalized Pythagoras trees with the goal to benefit from a multiple coordinated view. By this, a hierarchical data analyst can have a look into the data from different views to get more analytical support when exploring the data than if it was only presented in a single view. To make the tool accessible to researchers from all over the world we provide a web-based solution by using the Bokeh library based on the Python programming language. This supports the interactive visualization of data in a browser and hence, has the advantage of not being restricted to a certain kind of environment. We illustrate the usefulness of our visualization tool by means of applying it to the NCBI taxonomy that consists of more than 300,000 hierarchically organized species. Finally, we discuss scalability issues and limitations of our approach.

Index Terms—Hierarchy visualization, Interaction techniques, Multiple coordinated views, Web-based visualization

1 INTRODUCTION

In this paper we describe two visualization techniques for hierarchical data. To benefit from both of them we combine them in a multiple coordinated views representation [?] to not let alone the data analyst with a single perspective on the data. We implemented the generalized Pythagoras trees [?, ?] and the bubble hierarchies [?] technique that, on the one hand, create beautifully looking tree diagrams and, on the other hand, support the exploration of hierarchical data. Although there are many hierarchy visualizations available [?, ?] we base our work on those two concepts (see Figure ??).

To make the visualization tool accessible to a wide variety of users, we implemented it by using the Bokeh library which is Python-based and supports an easy way of providing interactive graphics in a web browser. By this strategy, hierarchical data analysts can upload their data, and see a visual representation of it in several views. The supported data format is the so called Newick format [?], i.e., our tool applies a parsing function to first read the data, before visualizing it.

To illustrate the usefulness of our visualization tool we apply it to the NCBI taxonomy dataset that consists of more than 300,000 hierarchically structured species. Our major finding with our combined visualization is that the NCBI taxonomy is a rather unbalanced hierarchical structure at a maximum depth of 42. Viruses, bacteria, and other organisms build the major subhierarchies, while the hierarchically deepest species are typically those living in the ocean, meaning that more species variety is given in this deeply nested hierarchical structures.

2 RELATED WORK

The visualization of hierarchical data is a central information visualization problem that has been studied for many years. Typical representations include node-link, stacking, nesting, indentation, or fractal concepts as surveyed by [?, ?]. Many variants of the general concepts exist, for instance, radial [?, ?] and bubble layouts [?, ?] of node-link diagrams, circular approaches for stacking techniques [?, ?, ?], or nested visualizations based on Voronoi diagrams [?, ?].

Although many tree visualizations were proposed in the past, none provides a generally applicable solution and solves all related issues.

For example, node-link diagrams clearly show the hierarchical structure by using explicit links in a crossing-free layout. However, by showing the node-link diagram in the traditional fashion with the root vertex on top and leaves at the bottom, much screen space stays unused at the top while leaves densely agglomerate at the bottom. Transforming the layout into a radial one distributes the nodes more evenly, but makes comparisons of subtrees more difficult. Node-link layouts of hierarchies have been studied in greater detail, for instance, [?] investigated visual task solution strategies whereas [?] analyzed space-efficiency.

Indented representations of hierarchies are well-known from explorable lists of files in file browsers. Recently, [?] investigated a variant as a technique for representing large hierarchies as an overview representation. Such a diagram scales to very large and deep hierarchies and still shows the hierarchical organization but not as clear as in node-link diagrams. Layered icicle plots [?], in contrast, use the concept of stacking: the root vertex is placed on top and, analogous to node-link diagrams, consumes much horizontal space that is as large as all child nodes together.

Treemaps [?], a space-filling approach, are a prominent representative of nesting techniques for encoding hierarchies. While properties of leaf nodes can be easily observed, a limitation becomes apparent when one tries to explore the hierarchical structure because it is difficult to retrieve the exact hierarchical information from deeply nested boxes: representatives of inner vertices are (nearly) completely covered by descendants. Treemaps have been extended to other layout techniques such as Voronoi diagrams [?, ?] producing aesthetic diagrams that, however, suffer from high runtime complexity.

Also, 3D approaches have been investigated, for instance, in Cone Trees [?], each hierarchy vertex is visually encoded as a cone with the apex placed on the circle circumference of the parent. Occlusion problems occur that are solved by interactive features such as rotation. Botanical Trees [?], a further 3D approach, imitate the aesthetics of natural trees but are restricted to binary hierarchies, that is, n -ary hierarchies are modeled as binary trees by the strand model of [?]; it becomes harder to detect the parent of a node.

The term fractal was coined by [?] and the class of those approaches has also been used for hierarchy visualization due to their self-similarity property [?, ?]. With OneZoom [?], the authors propose a fractal-based technique for visualizing phylogenetic trees; however, n -ary branches need to be visually translated into binary splits. [?] visualize random binary hierarchies with a fractal approach as botanical trees; no additional metric value for the vertices is taken into account; instead, they investigate the Horton-Strahler number for computing the branch thicknesses.

The goal of our work is to extend a fractal approach, which is closer to natural tree structures, towards information visualization. This goal promises embedding the idea of self-similarity and aesthetics of fractals into hierarchy visualization. Central prerequisite—and in this, our approach differs from existing fractal approaches—is that n -ary branches should be possible. With respect to information visualization, the approach targets at combining advantages of several existing techniques: a readable and scalable representation, an efficient use of screen space, and the flexibility for encoding additional information. To this end, we exploit the generalized Pythagoras trees [?, ?] and the bubble hierarchies [?] in a combined multiple coordinated views representation.

3 DATA MODEL

- What is hierarchical data in general?
- Describe the Newick format
- Explain the Newick format here with the benefits, but also with the drawbacks
- How easy is it to read the data into the tool and which parsing functionality or library has been used?

4 THE VISUALIZATION TOOL

We provide a visualization tool composed of several different views. To this end we support the generalized Pythagoras trees and the bubble hierarchies.

4.1 Graphical User Interface

- Show here the GUI of your tool. Make a screenshot from a browser and show how the hierarchy visualizations look like. Just an example screenshot...
- Explain all the features of the GUI in detail
- What are the different views of the GUI?
- Are they already linked?

4.2 Visualization Techniques

- Explain your visualization techniques here
- You can use the literature from www.treevis.net to get all the information you need
- Provide lots of screenshots from your visualization techniques here to illustrate them

4.3 Implementation Details

- how is the data read
- which programming language?
- which libraries, APIs?
- Is there an UML diagram for showing the design of the software architecture? If so, show it!!!
- Why did you choose this programming language? Which benefits are there?

5 APPLICATION EXAMPLE

- Show one dataset example, maybe the NCBI taxonomy in a browser
- Explain what you see
- Are there any visible structures?
- Are there any outliers or anomalies?

6 DISCUSSION AND LIMITATIONS

- What are the limitations of your approach?
- Is there a visual scalability issues like how many elements can be displayed on screen?
- Is there an algorithmic limitations, for example, for computing the hierarchy visualization and the layout?

7 CONCLUSION AND FUTURE WORK

- summarize your work in a few sentences
- what do you plan next? Interactions, more visualizations, make it faster?
- Evaluation by other users?
 - extension to other data formats
 - more interaction techniques
- further hierarchy visualizations

ACKNOWLEDGMENTS

We would like to thank our group members for their support in this DBL project. Moreover, we are thankful to our tutor who gave us a lot of feedback to find the right decisions in this project.