BioMaps: Towards Multiscale Views of Human Anatomy

Bernard de Bono[2], Michiel Helvenstijn[1], Joost Kok[1], and Natallia Kokash[1,⋆]

[1] Leiden Institute of Advanced Computer Science (LIACS), Leiden, The Netherlands
[2] Imperial College London

Abstract.

1 Introduction

In this paper, we present an integrated web-based environment for visualizing, browsing and editing a vast range of biomedical resources. The data is extracted from extensive publicly available taxonomies created and maintained by biomedical communities, pre-processed and arranged in a way that simplifies its perception and analysis of related biomedical processes.

First, we overview biomedical data resources we aim at providing access to and outline several usage scenarios that motivate our work (Section 2). Then we discuss data representation and visualization methods used to arrange and display relevant resources (Section 3). We describe a graphical tool prototype (Section 4). Furthermore, we overview related efforts and techniques (Section 5). Finally, we conclude the paper and discuss future work (Section 6).

2 Data resources and use scenarios

3 Visualizing taxonomies and connectivity data

Treemaps [1] are an effective technique to visualize hierarchical data by using nested shapes in a space-filling layout. Each shape represents a geometric region, which can be subdivided recursively into smaller regions. The standard shape is the rectangle. Nodes in a treemap, also called *tiles*, represent individual data items in a dataset. Node size, color and text label can be used to represent attributes of the data item. One-layered treemaps can display data attributes but are not very good at emphasizing the place of an item in the overall hierarchical structure. To compensate for that, a small margin with structural labels is typically used. In treemaps displaying hierarchical structures, it is possible to navigate among different layers and zoom into selected tiles [2].

To create a treemap, one must define a tiling algorithm - a way to divide a tile into sub-tiles of specified areas. Tiling algorithms used for typical applications of treemaps such as e.g., visualization of folders in files in the computer

[⋆] Corresponding author, email: `Natallia.Kokash@cwi.nl` file system with their respected sizes, do not associate tile positions with any characteristic of the data. This is not the case in our scenario: while a user navigates among different layers, filters data and zooms into selected areas, the tool should keep the tiles associated with the data in the same relative positions to each other. Otherwise, the user's perception of the displayed information will be quickly affected. Moreover, our tiling algorithm should allow the user to enforce constraints on tile positions to make the treemap views structurally resemble body regions. Hence, we developed a stable and customizable tiling algorithm that arranges tiles corresponding to data items according to a given template [].

For a set of n data items with no positional constraints, a default template is created that consists of $\lfloor \sqrt(n) \rfloor$ rows and $\lceil sqrt(n) \rceil$ columns in each row but the last one (which may contain less columns). If the positional data is available (e.g., FMA ontology adjacent-to relation) or a user wants to rearrange the data manually, a custom template is associated with the parent node of the dataset items. The template is a hierarchical structure $\{splitType, \{\}, ..., \{\}\}$ where $splitType \in \{slice, dice\}$ defines a way to split the rectangle into sub-rectangles: vertically or horizontally. By recursively splitting the available area into sub-rectangles, one can define complex layouts that enforce two dimensional constraints in the form "x is left/right of y" or "x is above/below of y" where x and y are individual data items or groups of data items that in their turn can be allocated as needed using the same technique.

4 Tool prototype

5 Related work

de Bono et al. [3] describes limitations of existing treemapping tools for biomedical data visualization. In [] we introduced a generic method to build custom templates which was applied in this paper to control layout of ApiNATOMY body tissues.

Cardiovascular GO Annotation Initiative `http://www.geneontology.org/GO.cardio.shtml`

3D Multiscale Physiological Human `http://www.springer.com/computer/image+processing/book/978-1-4471-6274-2`

Burch and Diehl [4] discuss the ways to display multiple hierarchies and conclude that overlaying connectors on top of treemaps is the most visually attractive and easy to follow approach. Among the alternative options they considered are separate, linked and colored tree diagrams, sorted and unsorted matrices and sorted parallel coordinate views. Regarding the way to layout the connectors, two naive methods were considered: straight connections and orthogonal connections.

Our application requires multiple taxonomies consisting of thousands of items to be displayed on relatively small screens of handhold devices. We employ the same visualization technique with more advanced treemapping and connector layout algorithms. Due to the large amount of vascular connectivity data, we employ hierarchical edge bundling method [5] to get intuitive and realistic representation of blood flow across a teemap-based plan of human body. In contrast to the scenarios in the aforementioned work, not every node in our vascular connection dataset has a corresponding node in the treemap. Thus, force-directed graph drawing method [] is added to the scene to find optimal positions of intermediate nodes on the paths that connect the root of the taxonomy (heart) with its leaves (body tissues shown as treemap tiles). The variation of the force-bundling method suitable for our application is known as sticky force-directed bundling [] which allows to fix the positions of certain nodes and allocate other nodes to achieve mechanical equilibrium between forces pulling the free nodes towards fixed positions.

Other potentially useful methods did not provide the desired result. The first approach we tried consists of applying the force-directed edge bundling method [6] to bundle entire paths among the heart chambers and body tissues does not reflect the hierarchical structure of vascular connectivity graph. The second approach, force-based edge bundling over a graph produced by sticky force-directed node allocation algorithm results into unnatural distortion of short edges towards each other. Other edge-bundling methods [] operate on graphs with known node positions and thus would produce visualizations on our data that suffer from similar problems.

6 Conclusions and Future Work

References

1. Johnson, B., Shneiderman, B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: Proc. of the 2nd Conference on Visualization ’91, IEEE (1991) 284–291
2. Blanch, R., Lecolinet, E.: Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. TVCG **13** (2007) 1248–1253
3. de Bono, B., Grenon, P., Sammut, S.: Apinatomy: A novel toolkit for visualizing multiscale anatomy schematics with phenotype-related information. Hum Mutat **33**(5) (2012) 837–848
4. M.Burch, S.: Trees in a treemap: Visualizing multiple hierarchies. In: Proc. VDA 2006. (2006)
5. Holten, D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. IEEE Transactions on Visualization and Computer Graphics **12**(5) (2006) 741–748
6. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. Comput. Graph. Forum **28**(3) (2009) 983–990