

ApiNATOMY: Towards Multiscale Views of Human Anatomy

Bernard de Bono², Pierre Grenon², Michiel Helvenstijn¹, Joost Kok¹, and
Natallia Kokash^{1,*}

¹ Leiden Institute of Advanced Computer Science (LIACS), Leiden, The Netherlands

² University College London (UCL)

Abstract.

1 Introduction

An ontology represents a set of concepts and their interrelations in a specific domain. Ontologies are created and maintained by domain experts and used as reference taxonomies by other professionals as well as software tools supporting their activities. The need for effective ontology visualization for design, management, and browsing has arisen. Consequently, many ontology visualization tools have been developed to assist semantic knowledge acquisition and maintenance [1]. However, the major part of this knowledge remains inaccessible for people without expertise in semantic web technologies and understanding of design principles for specific types of ontologies. We aim at addressing this issue in biomedical domain by generating body plans and schematics for ongoing biomedical processes from existing ontological resources.

In this paper, we present an integrated web-based environment for biomedical experts to navigate through schematically represented body tissues, browse, analyze and extend associated data resources and reason about multi-scale processes in selected regions of human body at any level of detail: from body parts and organs, to cells, molecules, proteins, and gene expressions. More specifically, we present a tool prototype that visualizes body parts, cardiovascular and neural connections, mathematical models of ongoing processes, static and dynamic 3D models, protein structures,...

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).

* Corresponding author, email: nkokash@liacs.nl

2 Data resources and use scenarios

FMA - taxonomy, connectivity data

Genes ontology

CellML

3D models

We arrange data in a hierarchical way starting from the large-scale views showing external and internal surfaces and organs arranged to resemble the longitudinal section through the middle of the human body as the top-level taxonomy. Figure 1 shows the idealized radially symmetric body plan, apportioned over cylindrical regions. This homunculus has a central longitudinal axis of rotation located in the idealized gut lumen which runs in the cephalocaudal direction. Each of the organs in the plan is composed of multiple tissues and sub-organs, the structural information about them is obtained from the Foundational Model of Anatomy (FMA) ontology³. As the visualization of the full ontology may obscure the details a user is interested to see, it is essential for the visualization tool to support data filtering across multiple levels and contextual zooming into selected areas. The user should be able to create a custom view with the internal structure of the selected body parts placed in a way that simplifies the analysis of these data.

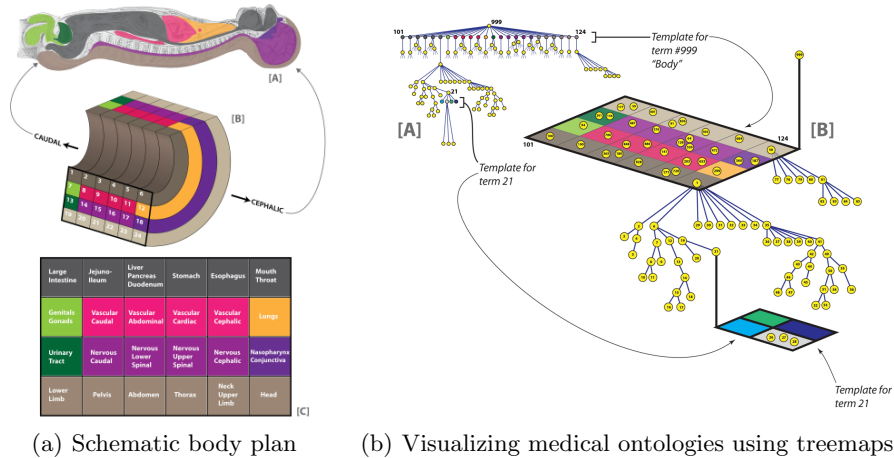


Fig. 1. Longitudinal section through the middle of the male human body

³ Foundational Model of Anatomy - <http://sig.biostr.washington.edu/projects/fm/>

3 Visualizing taxonomies and connectivity data

Treemaps [2] 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 [3].

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 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 [4].

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 $\{splitType, \{\}, \dots, \{\}\}$ 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.

3.1 Connectivity data

We use treemap-based body plans as background to overlay the schematic representation of biological systems such as circulatory, respiratory, or nervous systems. Body systems are essentially graphs or hypergraphs with nodes corresponding to body parts (treemap tiles), entities that belong to body parts (proteins, cells, etc.) or auxiliary nodes that are not displayed on the treemap but still carry important biomedical information, while their edges or paths rep-

resent organ system compounds such as blood vessels or nervous connections that pass through the certain body parts or its sub-parts.

Body systems are intrinsically complex and require efficient data visualization techniques that would help us to avoid clutters induced by the large amount of graph edges and their crossings and allow users to overview large parts of the body systems as well as to trace individual connections and analyze their structure. Edge bundling techniques have been proposed to improve perception of connectivity data in large dense graphs. Such techniques generally rely on edge rerouting strategies that are either solely target at improving visual perception of edges constrained by positions of nodes or exploit the relations among connectivity data as guidelines for more natural allocation of graph edges and nodes. Our application requires the mixture of these techniques

As example, we consider the schematic visualization of blood vessels in human body. The initial dataset is a graph extracted from the FMA database which consists of approximately 11,300 edges and over 10,000 distinct nodes. An edge represents an unbranched segment of a blood vessel. Nodes represent junctions between blood vessel segments. Samples of records from the dataset are shown in Table 1. The first column in the dataset is a unique vascular segment identifier (ID). The second column bears the vessel type, i.e. any one of four numbers that give an indication of the biological type of segment: 1 - for arterial segment, 2 - microcirculation (MC), there are three types of MC edges: arteriole, capillary and venule, 3 - venous, and 4 - cardiac chamber. The third column bears FMA IDs. For non-MC edges (i.e. vessels of type 1, 3 or 4) the number in this column is the FMA ID of the blood vessel of which that segment is part (e.g. there are over 50 segments/edges that form part of the trunk of the aorta, i.e., FMA ID 50010931). For MC edges (type 2), the FMA ID is that of the body region in which the MC is embedded. The fourth and fifth columns in the dataset bear the unique node identifiers in an edge pair. The sixth column is a free-text label describing that segment (i.e. edge).

When it comes to describing routes of blood passage through the heart, one has to distinguish between cardiac chambers, (i.e. main lumen of left ventricle, left atrium, right ventricle or right atrium) and the MC of the wall of heart (e.g. wall of left ventricle (FMA ID 7101), left atrium (7097), right ventricle (7098) or right atrium (7096)). There are two ways by which blood passes through the heart: (i) though the chambers and (ii) through its walls. Consequently, we exclude cardiac chamber segments from consideration as they are redundant for the representation of vascular connectivity data in our application. An MC is represented by three edges connected in series: one edge represents tissue arterioles, a second edge stands for the bed of capillaries, while a third denotes the venules. In one MC, therefore: a) the end node of the arteriolar edge and the start node of the capillary edge are equivalent, and b) the end node of the capillary edge and the end node of the venular edge are equivalent.

In the above example, the anatomical entity in which the MC is embedded is 62528 - the topology of MC segment connectivity is as follows:

$$62528.2 - [121a] \rightarrow 62528.4 - [121c] \rightarrow 62528.5 \leftarrow [121v] - 62528.3.$$

Table 1. Vascular connectivity data from the FMA ontology

Segment	Type	FMA	Node 1	Node 2	Description
121a	2	62528	62528_2	62528_4	Arterioles in Microcirculation segment of Wall of left inferior lobar bronchus
121c	2	62528	62528_4	62528_5	Capillaries in Microcirculation segment of Wall of left inferior lobar bronchus
121v	2	62528	62528_3	62528_5	Venules in Microcirculation segment of Wall of left inferior lobar bronchus
...
8499	1	69333	8498_0	62528_2	Arterial Segment 8499 of Trunk of left second bronchial artery from origin of supplying terminal segment to the arteriolar side of the Wall of left inferior lobar bronchus MC
9547	3	66699	9546_0	62528_3	Venous Segment 9547 of Trunk of left bronchial vein from origin of supplying terminal segment to the venular side of the Wall of left inferior lobar bronchus MC

MC segment 121a is supplied with blood by the arterial segment 8499 while MC segment 121v is drained of blood by the venous segment 9547.

The accurate visualization of the cardiovascular system in a comprehensible way requires complex pre-processing (about 12 rules were identified to extract the data of interest from the presented dataset by a biomedical expert in our team). In this paper, for the illustration purpose we show only paths connecting MCs of the walls of the heart to MCs belonging to the sub-organs of the organs in our 24 upper level tile body plan. To obtain this view, we looked for the shortest paths (due to the way the data is represented in the initial data set, loops are possible) from the MCs of the heart walls to the final FMA tiles. For example, the path from the left ventricle to the wall of left inferior lobar bronchus MC is as follows: 7101 \rightarrow 2406 \rightarrow 2407 \rightarrow 2573 \rightarrow 2657 \rightarrow 3780 \rightarrow 4288 \rightarrow 4707 \rightarrow 4709 \rightarrow 4711 \rightarrow 4713 \rightarrow 4714 \rightarrow 8492 \rightarrow 8494 \rightarrow 8496 \rightarrow 8498 \rightarrow 8499 \rightarrow 62528. while the path from this organ to the right atrium is 7096 \leftarrow 771 \leftarrow 1333 \rightarrow 1334 \leftarrow 1731 \leftarrow 1993 \leftarrow 1994 \leftarrow 1996 \leftarrow 1998 \leftarrow 1999 \leftarrow 1983 \leftarrow 9534 \leftarrow 9536 \leftarrow 9538 \leftarrow 9540 \leftarrow 9542 \leftarrow 9544 \leftarrow 9546 \leftarrow 9547 \leftarrow 62528.

The first and the last IDs in this path correspond to the tiles in the treemap view, while the intermediate IDs will be represented using auxiliary nodes with undefined coordinates. One of the issues we encountered is the need to determine optimal positions for these nodes. Since several paths as above can have common sub-paths, the intermediate nodes should not deviate too much from the way from the heart MC to each of the end tiles sharing such sub-paths. This motivates our application of the sticky force-directed graph visualization method that given a sub-set of nodes with fixed coordinates arrange other nodes to optimize forces on the graph edges. As our only objective is to allocate intermediate nodes as close to the paths they belong to (which most naturally can be represented by straight lines) as possible, we set gravity and charge to 0 and the desired

link length to a minimal possible length. The extracted connectivity graphs for two heart walls: left ventricle (7101) and right atrium (7096), are shown in Figures 2(a) and 2(b).

Once the coordinates for all connectivity graph nodes are determined, we apply hierarchical edge bundling method that uses the path structure to bundle paths with common sub-paths in the graph obtained on the first stage. The final set of paths from left ventricle and right atrium are shown in Figures 2(c) and 2(d), respectively. In these images, both left ventricle and right atrium are descendants of tile 6000009: Vascular Cardiac. By setting bundling algorithm tension parameter close to 1 (0.96-0.99), we obtain a view that clearly shows that individual paths extracted from our dataset form “highways” which correspond to major arteries and veins in the human body.

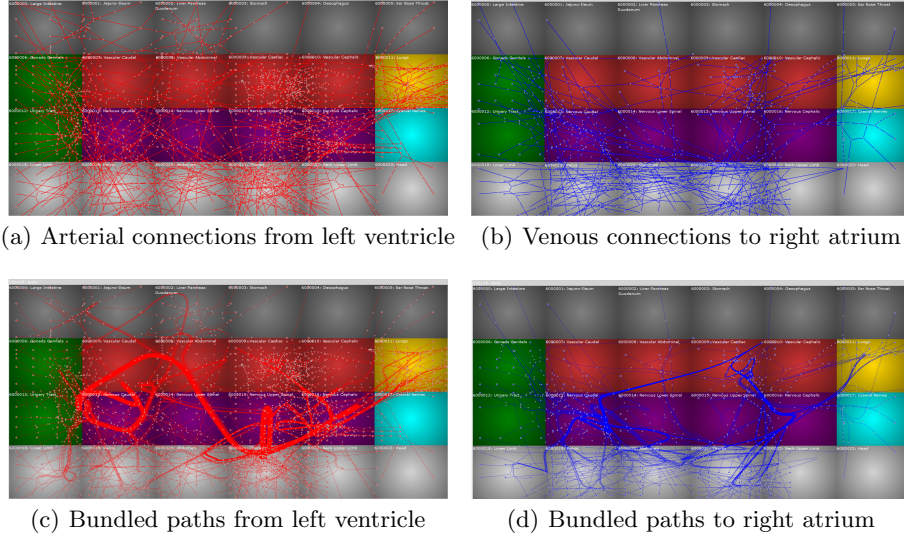


Fig. 2. Cardiovascular system

After one-time pre-processing to import data from available external sources, we store connectivity data in a generic JSON format. A user can interact with and edit these data using functionality of our application. One of the prime goals of our tools is to simplify the access and maintenance of biomedical taxonomies, vascular system among others. A user should be able to choose a way to represent connectivity data. There is ongoing work on the implementation of the orthogonal connector visualization algorithm that would place links into margins between tiles so that they will not obstruct the interaction with the tiles.

4 Tool prototype

- Architecture
- Description: views, interaction
- ...

5 Related work

de Bono et al. [5] describes limitations of existing treemapping tools for biomedical data visualization. In [1] 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 [6] 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 handheld 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 [7] 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 teemap. Thus, force-directed graph drawing method [8] 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 teemap tiles). The variation of the force-bundling method suitable for our application is known as sticky force-directed bundling [9] 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 [8] 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 [10] 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. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.: Ontology visualization methods—a survey. *ACM Comput. Surv.* **39**(4) (2007)
2. 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
3. Blanch, R., Lecolinet, E.: Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. *TVCG* **13** (2007) 1248–1253
4. N. Kokash, de Bono B., K.J.: Template-based treemaps to preserve spatial constraints. In: *Proc. IVAPP 2014*. (2014)
5. 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
6. M.Burch, S.: Trees in a treemap: Visualizing multiple hierarchies. In: *Proc. VDA 2006*. (2006)
7. Holten, D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics* **12**(5) (2006) 741–748
8. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. *Comput. Graph. Forum* **28**(3) (2009) 983–990