ApiNATOMY: Towards Multiscale Views of Human Anatomy

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

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

Abstract. Physiology experts deal with complex biophysical relationships, across multiple spatial and temporal scales. Automating the discovery of such relationships, in terms of physiological meaning, is a key goal to the physiology community. ApiNATOMY is an effort to provide an intuitive graphical interface for managing ontologies and semantic metadata relevant to physiology. In this paper, we present a web-based ApiNATOMY environment for physiology experts to navigate through circuitboard visualizations of body components and their connections across scales. In particular, we present a tool prototype that visualizes schematics of ontology-based knowledge about body parts and their cardiovascular and neural connections. Graphical renderings of gene products and mathematical models of processes that are semantically annotated with this knowledge are overlaid on these schematics.

1 Introduction

An ontology consists of a set of terms, and their relations, representing a specific domain of knowledge. They are created and maintained by knowledge domain experts, and are used as computer-readable taxonomies by software tools to support knowledge management activities in that domain.

The complexity of some of the ontologies in current use, as well as the complexity of handling semantic metadata that annotate third party resources with ontology terms (e.g. as described in [1]), has generated considerable demand for effective visualization in the design, authoring, navigation and management of (i) ontology-based knowledge and (ii) semantic metadata that make use of ontologies.

In response to the above demand, a number of generic ontology visualization tools have been developed to assist knowledge acquisition, browsing and maintenance of ontologies [2]. Such tools, however, put considerable and unrealistic demands on the users' familiarity and expertise in both (i) semantic web technologies and (ii) the design principles of ontologies. It is unlikely that a user with expertise in the *domain* of an ontology also has expertise in the *technologies* that handle ontologies.

⋆ Corresponding author, email: nkokash@liacs.nl

The domain of biomedical physiology is a case in point. Physiology experts deal with complex biophysical operations, across multiple spatial and temporal scales, which they represent in terms of the transfer of energy from one form to another and/or from one anatomical location to another. Different kinds of descriptions of these biophysical operations are produced by different disciplines in biomedicine. For instance, (i) a medical doctor may describe the mechanism by which a stone in the ureter causes damage in the kidney; (ii) a pharmacologist may depict the process by which a drug absorbed from gut transits to the hip joints where it reduces inflammation; (iii) a molecular geneticist may trace the anatomical distribution of the expression of particular gene to understand the cause of a skeletal malformation; and, (iv) a bioengineer may build a mathematical model to quantify the effect of hormone production by the small intestine on the production of bile by the liver. These descriptions take diverse forms, ranging from images and free text (e.g., a paper in a journal) to XML documents bearing well-defined data (e.g. from a clinical trial) or sets of model variables and related equations (which could be used for as input for a simulation tool).

Automating the discovery of relationships, in terms of physiological meaning, between the above types of description is a key goal to the physiology community. To that end, this community is investing considerable effort in building ontologies for the annotation and semantic management of resources describing physiology. A number of reference ontologies have been created to represent the various entities required to describe physiology, including gene products [3], chemical entities [4], cells [5] and gross anatomy [6]. Cumulatively, these reference ontologies consist of hundreds of thousands of terms, such that the volume of semantic metadata arising from annotation of resources with these terms is considerable. However, conventional technology for the management of ontologies and metadata is not usefully accessible to physiology experts.

The ApiNATOMY effort has emerged to provide intuitive graphical interface for managing ontologies and semantic metadata relevant to physiology. In this paper, we present a web-based ApiNATOMY environment for physiology experts to navigate through circuitboard visualizations of body components and their physiological connections across scales. In particular, we present a tool prototype that visualizes schematics of ontology-based knowledge about body parts and their cardiovascular and neural connections. Graphical renderings of semantic annotations to (i) gene product data and (ii) process models are overlaid onto these schematics, in support of biomedical knowledge management use cases discussed in the next Section.

This paper is structured as follows: first, we give an overview of the ontology, metadata and data resources that we focused on for this prototype, and outline key usage scenarios that motivate our work (Section 2). We then discuss representation and visualization methods applied to arrange and display relevant resources (Sections 3 and 4). In Section 5, we describe the graphical tool prototype that implements these methods. Furthermore, in Section 6, we overview related methodologies, efforts and techniques in the field. Finally, we conclude

the paper with a discussion of the anticipated implications of this tool, as well as and planned future work.

2 Use Case, Resources and Early-stage Results

In this section, we briefly discuss core use cases for the ApiNATOMY application: the generation of interactive schematics in support of genomics and drug discovery studies. In so doing, we introduce (i) some of the key ontological and data resources required in this case, as well as (ii) early-stage results of the ApiNATOMY application effort.

The domains of genomics and drug discovery are dependent on physiology knowledge, as both domains take into account the manufacture of gene products in different parts of the body and the regulated long-distance transport of molecules that interact with these products (e.g. drugs, nutrients, or gene products). Data about (i) the location of gene product manufacture (i.e. gene expression data, such as [7]), as well as (ii) transport routes taken by molecular interactors (e.g. as derived from from pharmacokinetic modeling resources such as [8]) may be usefully depicted in the form of a physiology *circuitboard*.

In ApiNATOMY, a physiology circuitboard schematic consists of a combination of (i) an anatomical treemap and (ii) an overlay of process graphs. In our earlier prototypes (described in [9,10]), templates were applied to constrain the layout of tiles in treemaps of the Foundational Model of Anatomy (FMA) [6] ontology, such that nesting of one tile inside another indicates that the term asscoiated with the child tile is either a mereotopological *part* or a *subclass* of the term associated with the parent tile.

In the following sections, we discuss our early-stage results in the construction of an ApiNATOMY Graphical User Interface (GUI) that supports user interaction with circuitboard schematics via point-and-click navigation of the treemap content. This type of interaction extends to also to involve process graphs. We arrange ontology-based anatomy data in a hierarchical structure, starting from the upper level views of external and internal surfaces and organs. This upper level is arranged to resemble the longitudinal section through the middle of the human body (Figure 1(b)). Each of the organs in the plan is composed of multiple tissues and sub-organs and, in the prototype presented here, the structural information about them is obtained from the FMA ontology [6]. The GUI tool also supports data filtering across multiple levels and contextual zooming into selected areas.

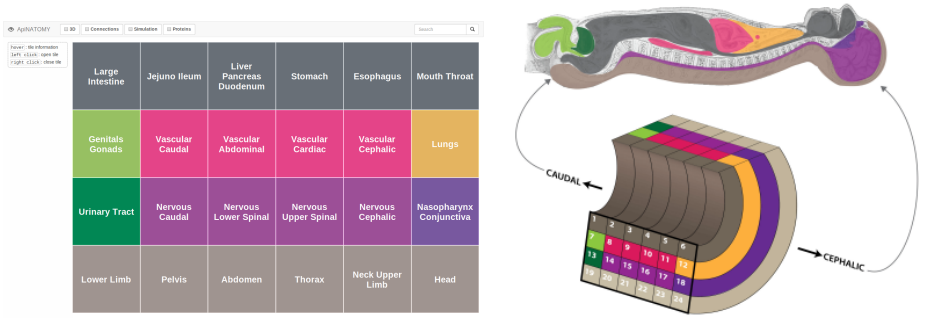
In addition, here we report on the graphical projection of routes of (i) blood flow processes linking different regions of the human body (using data generated in [11]), as well as (ii) transport processes along neurons of the central nervous system (i.e. brain and spinal cord) with data obtained via the Neuroscience Information Framework [12].

The ApiNATOMY GUI is built from inception as a three dimensional (3D) environment. This facilitates interaction not only with 3D renderings of the circuit boards themselves, but also with a wide range of geometry/mesh formats

for volumetric models of biological structure across scales. For instance, it is already possible to overlay Wavefront .obj data from BodyParts3D [13] as well as SWC data provided by neuromorpho.org [14]. The management of such data is critical to the understanding of long-range molecular processes in genomics and drug discovery research.

In the next two sections, we discuss our results in:

- constraining of treemap layouts to generate stable anatomical treemaps,
- designing and overlaying physiological communication routes for the cardiovascular and neural systems, and
- querying and depicting of protein architecture diagrams for the anatomical overview of gene expression data and mathematical models relevant to molecular mechanisms.



(a) Initial view of ApiNATOMY

(b) Longitudinal section through the male human body, justifying the layout

Fig. 1. The main 24-tile layout of the ApiNATOMY circuitboard

3 Visualizing Taxonomies and Connectivity Data

Treemaps [15] 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 a 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 [16].

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 tiles should be kept in the same relative positions to each other. Otherwise, the user's perception of the displayed information will be quickly disrupted. 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 according to a given template [10].

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 fewer 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 $\{splitType, \{\}, \ldots, \{\}\}$ where $splitType \in \{\text{slice}, \text{dice}\}$ 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.

The schematic body plans created using template-based treemaps can be seen in Figure 1. Figure 1(a) shows the top level 24 tile body anatomy plan. The choice of this layout is explained by Figure 1(b), which shows how it can conceptually wrap around the longitudinal axis of the human body. The treemap layout is controlled by the (default) templates and remains stable during navigation.

3.1 Connectivity data

With the treemap-based body plans as background, we overlay the schematic representation of *body systems* such as circulatory, respiratory, or nervous systems. Body systems are essentially graphs with nodes corresponding to body parts (treemap tiles) or entities inside of body parts (proteins, cells, etc.), and their edges represent organ system compounds such as blood vessels or nervous connections that pass through such body parts or sub-parts. They may also contain auxiliary nodes that are not represented on the treemap but still carry important biomedical information.

Body systems are intrinsically complex and require efficient data visualization techniques to help avoid clutter induced by the large amount of graph edges and their crossings. To support the GUI functionality, body systems must allow users to overview large parts of the body systems as well as to trace individual connections and analyze their structure. In this context, edge bundling techniques [17,18,19] 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 targeted at improving visual perception (by using the positions of nodes) or exploit the relationships among connectivity data as guidelines for a more natural allocation of graph edges and nodes. Our application requires a mixture of these techniques.

As an example, consider the schematic visualization of blood vessels in human body. Our initial dataset is a graph based on the FMA resoure, and consists of approximately 11,300 edges and over 10,000 distinct nodes. In this case, an edge represents a flow process over an unbranched segment of a blood vessel. Nodes represent blood vessel junctions and end-points. 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). 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).

(MH: Probably much of the information in the next two paragraphs could be cut.)

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

To describe routes of blood passage through the heart, it is important to distinguish between cardiac chambers, (i.e. main lumen of left ventricle, left atrium, right ventricle or right atrium) and the blood supply (i.e. MCs) to 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:

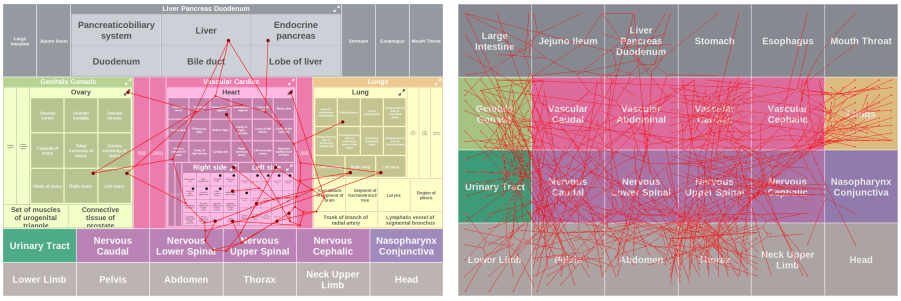
$$62528\_2 \xrightarrow{121a} 62528\_4 \xrightarrow{121c} 62528\_5 \xleftarrow{121v} 62528\_3.$$

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 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 upper level 24 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, cycles 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 looks like $7101 \rightarrow 2406 \rightarrow \cdots \rightarrow 8499 \rightarrow 62528$, while the path from this organ to the right atrium is like follows: $7096 \leftarrow 771 \leftarrow \cdots \leftarrow 9546 \leftarrow 9547 \leftarrow 62528$.

The first and the last IDs in this path correspond to the tiles in the treemap, 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 [20,21] in which a sub-set of nodes have fixed coordinates, and the coordinates of the other nodes is determined by simulating imaginary forces applied by their edges (Figure 2(a)).

If there are too many edges to get a clear overview of the data —as in Figure 2(b), which shows the full connectivity graph for the left ventricle (7101) on the top-level body plan— we can apply a hierarchical edge bundling method that uses the path structure to bundle common sub-paths. The result for the left ventricle is shown in Figure 2(c), which gives a much nicer overview. The result for the right atrium is shown in Figure 2(d).



(a) Selected blood vessel connections (b) Arterial connections from left ventricle

(c) Bundled paths from left ventricle (d) Bundled paths to right atrium

Fig. 2. Cardiovascular system

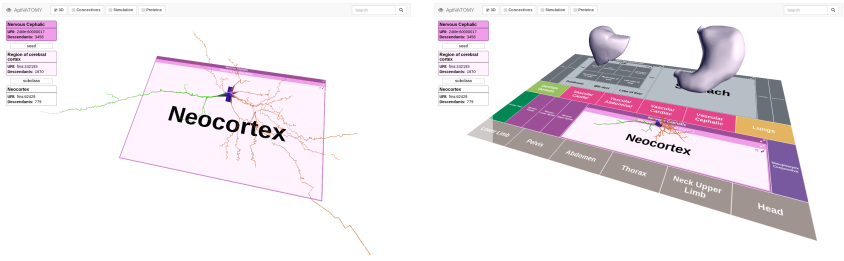
After a one-time pre-processing to import data from available external sources, we store connectivity data in a convenient format. A user can interact with and edit this data using our application.

One of the prime goals of the tool is to simplify the access and maintenance of biomedical taxonomies, including those concerned with physiological connectivity data. A user should be able to customize the way in which that data is represented. There is ongoing work on an implementation of the orthogonal connector visualization algorithm that would place edges into margins between tiles so that they will not obstruct the interaction with the tiles themselves.

4 Visualization of Models and Metadata

The entities in the ApiNATOMY ontologies have various data associated with them, to which they are explicitly linked via semantic metadata annotations. This includes, for instance, static and dynamic 3D models of body organs and their subsystems. To illustrate the application of ApiNATOMY in the management of semantic metadata and associated resources, we extract and display neuronal reconstructions and associated metadata from `http://neuromorpho.org` [14]. Figure 3(a) shows a sample neuron model associated with the neocortex (reached through"Nervous Cephalic" → "Region of cerebral cortex" → "Neocortex"). ApiNATOMY allows users to show multiple 3D objects together

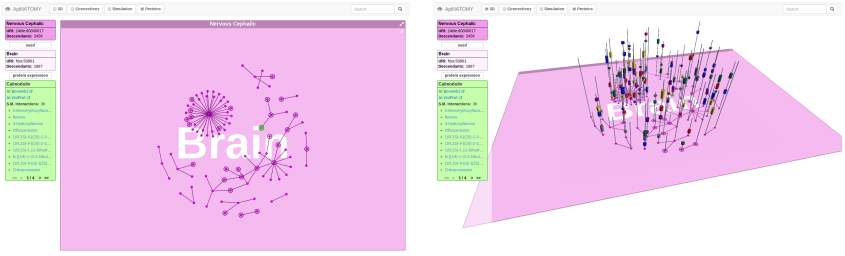
in their proper context. For example, Figure 3(b) shows a screenshot of a circuit board with the "Neocortex" neuron, as well as 3D models of the "Liver" and "Stomach".



(a) Neuron in the "Neocortex" (b) Multiple 3D models in context

Fig. 3. Visualization of static 3D models

ApiNATOMY also supports the visualization of protein- and drug-interaction networks (Figure 4(a)) that are represented as graphs on top of treemap tiles. We are in the process of acquiring and integrating relevant data from the Ensembl genomic database http://www.ensembl.org/. In Ensembl, gene models are annotated automatically using biological sequences data (e.g. protein, mRNA). We query this database to extract genes, transcripts, and translations with related protein features such as e.g., PFAM domains, and locate automatically-generated diagrams of proteins with FMA tiles to represent the anatomical location in which they are expressed. For ease of visualisation, protein diagrams show domain features built from different shapes and colors in a navigable 3D environment (Figure 4(b)). (MH: cite?)



(a) Protein interactions in 2D (b) Protein features in 3D

Fig. 4. Visualizing protein expression, protein interaction and protein features

5 Implementation

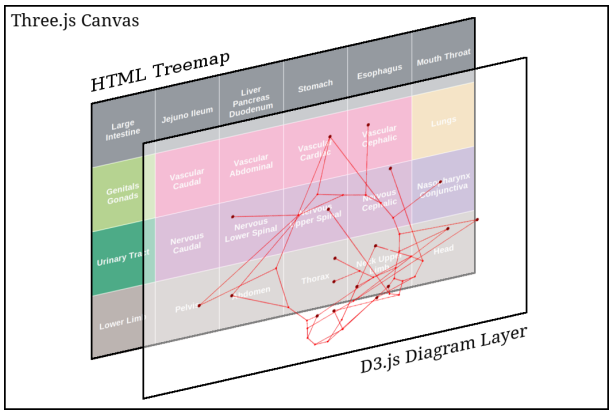


Fig. 5. The three layers of circuitboard visualization.

In this section we discuss a number of implementation aspects of ApiNATOMY. For maximum compatibility across operating systems as well as handheld devices, the whole application is written in Javascript. The main framework in use is AngularJS, which provides a Model-View-Controller architecture, as well as two-way databinding. Connectivity- and protein-protein interaction diagrams (Figures 2 and 4(a)) are generated using D3.js, and all 3D functionality (Figures 3 and 4(b)) is implemented using Three.js, which provides a convenient abstraction layer over WebGL.

5.1 Circuitboard Layers

The circuit-board is rendered with essentially three layers, which are shown in Figure 5. The treemap is generated with plain HTML. On top of this, a partly transparent diagram layer is rendered by D3.js. The positions of the tiles and the positions of the diagram nodes are synchronized with AngularJS two-way databinding. When 3D mode is activated, Three.js takes control of both layers. Besides rendering 3D objects with WebGL, it can manipulate HTML elements using CSS 3D transforms. When using both rendering engines in conjunction, Three.js can keep WebGL and HTML perfectly synchronized. Together with AngularJS two-way databinding, we get very fine control of positioning. This is demonstrated particularly well by Figure 4(b).

5.2 Focus and Direct Feedback

A separate module keeps track of the entity under focus. Whenever the mouse hovers over a specific tile or object, it is highlighted and its hierarchical informa-

tion is shown in the left side-panel (Figures 3 and 4). Clicking on the object fixes this focus, allowing the user to interact with the information in the side-panel.

This direct feedback has another purpose. An ontology need not necessarily be a tree. In the FMA ontology, for example, different branches may join, making it a directed acyclic graph. A treemap, however, by its very nature, is only suitable for visualizing trees. We compensate for this by allowing the same entity to be represented by more than one tile at the same time. To reinforce this intuition, all such tiles are highlighted in unison when the mouse hovers over any one of them. Only one tile per entity is considered 'active'. Only the active tile may be opened up to show its children, and only the active tile participates in the visualization of cross-tile connectivity data.

6 Rationale for our approach and related work in anatomy and physiology knowledge visualization

The need for the multi-scale visualization and analysis of human body systems is well recognized by biomedical communities. For example, the 3D Multiscale Physiological Human initiative deals with combinations of physiological knowledge and computational approaches to help scientists in biomedicine to improve diagnostics and treatments of various disorders [22,23]. In addition, numerous anatomy-related taxonomies and databases have been created and are widely used by researchers in the biomedical field [24]. While various generic visualization techniques can be used to display biomedical ontologies [2], to the best of our knowledge, ApiNATOMY is the first systematic approach to integrate such knowledge in one extensible and configurable framework.

Among the most effective taxonomy visualization techniques are space-filling diagrams, and in particular, treemaps. de Bono et al. [9] describes limitations of existing treemapping tools for biomedical data visualization. To overcome these limitations, we introduced a generic method to build custom templates which is applied in our tool to control layout of ApiNATOMY body tissues. Among the advantages of the proposed treemapping method are customizable layouts, visualization stability and multi-focus contextual zoom. The detailed comparison of our method with existing treemaping algorithms can be found in [10]. Burch and Diehl [25] 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 potentially large amount of vascular connectivity data, we employ the hierarchical edge bundling technique [17] that results in an intuitive and realistic depiction of blood flow across a treemap-based plan of

the 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 [26] is added to the scene to find optimal positions of intermediate junctions on the paths that connect the root of the taxonomy (i.e. in the 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 placement [20] 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 [27] to bundle entire paths among the heart chambers and body tissues, but this 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(e.g., [18,19,28]) operate on graphs with known node positions and thus would produce visualizations on our data that suffer from similar problems.

7 Conclusions and Future Work

References

1. de Bono, B., Hoehndorf, R., Wimalaratne, S., Gkoutos, G., Grenon, P.: The RICORDO approach to semantic interoperability for biomedical data and models: strategy, standards and solutions. BMC Research Notes **4** (2011) 313
2. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.: Ontology visualization methods - a survey. ACM Comput. Surv. **39**(4) (2007)
3. Blake, J.A., et al.: Gene ontology annotations and resources. Nucleic Acids Res. **41** (2013) D530–535
4. Hastings, J., de Matos, P., Dekker, A., Ennis, M., Harsha, B., Kale, N., Muthukrishnan, V., Owen, G., Turner, S., Williams, M., Steinbeck, C.: The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013. Nucleic Acids Res. **41**(D1) (2013) D456–463
5. Bard, J., Rhee, S.Y., Ashburner, M.: An ontology for cell types. Genome Biol. **6**(2) (2005) R21
6. Rosse, C., Mejino(Jr), J.L.V.: A reference ontology for biomedical informatics: the foundational model of anatomy. J. Biomed. Inform. **36**(6) (2003) 478–500
7. EBI: Arrayexpress home, EBI. web site (2012)
8. L. Harnisch, I. Matthews, J.C., Karlsson, M.O.: Drug and disease model resources: a consortium to create standards and tools to enhance model-based drug development. CPT Pharmacomet. Syst. Pharmacol. **2** (2013) e34
9. 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
10. Kokash, N., de Bono B., J., K.: Template-based treemaps to preserve spatial constraints. In: Proc. IVAPP 2014. (2014)

11. de Bono, B.: Achieving semantic interoperability between physiology models and clinical data. In: Proc. of IEEE Int. Conf. on e-Science Workshops. (2011) 135–142
12. Gardner, D., et al.: The neuroscience information framework: A data and knowledge environment for neuroscience. Neuroinformatics **6**(3) (2008) 149–160
13. Mitsuhashi, N., Fujieda, K., Tamura, T., Kawamoto, S., Takagi, T., Okubo, K.: Bodyparts3d: 3d structure database for anatomical concepts. Nucleic Acids Res. (**37**) D782–D785
14. Ascoli, G.A.: Mobilizing the base of neuroscience data: the case of neuronal morphologies. Nat. Rev. Neurosci. **7**(4) (2006) 318–324
15. 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
16. Blanch, R., Lecolinet, E.: Browsing zoomable treemaps: Structure-aware multiscale navigation techniques. TVCG **13** (2007) 1248–1253
17. Holten, D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. IEEE Transactions on Visualization and Computer Graphics **12**(5) (2006) 741–748
18. Gansner, E.R., Hu, Y., North, S.C., Scheidegger, C.E.: Multilevel agglomerative edge bundling for visualizing large graphs. In Battista, G.D., Fekete, J.D., Qu, H., eds.: Proce of PacificVis, IEEE Computer Society (2011) 187–194
19. Hurter, C., Ersoy, O., Telea, A.: Graph bundling by kernel density estimation. Comp. Graph. Forum **31** (2012) 865–874
20. Fruchterman, T., Reingold, E.: Graph drawing by force-directed placement. Software Practice and Experience **21**(11) (1991) 1129–1164
21. Bostock, M.: Sticky force layout. Online visualization tool (accessed on 09-14-2014)
22. Magnenat-Thalmann, N., Ratib, O., Choi, H.F., eds.: 3D Multiscale Physiological Human, Springer (2014)
23. Magnenat-Thalmann, N., ed.: Modelling the Physiological Human: Proceedings of the Second 3D Physiological Human Workshop. Volume 5903 of LNCS., Springer (2009)
24. Burger, A., Davidson, D., Baldock, R., eds.: Anatomy Ontologies for Bioinformatics. Volume 6 of Computational Biology., Springer (2008)
25. Burch, M., Diehl, S.: Trees in a treemap: Visualizing multiple hierarchies. In: Proc. VDA 2006. (2006)
26. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall (1999)
27. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. Comput. Graph. Forum **28**(3) (2009) 983–990
28. Selassie, D., Heller, B., Heer, J.: Divided edge bundling for directional network data. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis) (2011)