ApiNATOMY: Towards Multiscale Views
of Human Anatomy

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

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

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 interface between the physiology expert's knowledge and all ranges of data relevant to physiology. It does this through an intuitive graphical interface for managing semantic metadata and ontologies relevant to physiology. In this paper, we present a web-based ApiNATOMY environment, allowing physiology experts to navigate through circuitboard visualizations of body components, and their cardiovascular and neural connections, across different scales. Overlaid on these schematics are graphical renderings of organs, neurons and gene products, as well as mathematical models of processes semantically annotated with this knowledge.

1   Introduction

Knowledge of physiology is extensive and complex. To provide software support for using and manipulating physiology data, formalization of the knowledge is required. 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. When knowledge is formalised in this way, it is possible to record explicit descriptions of data elements in the relevant domain using ontologies; this is the process of semantic annotation or the generation of *semantic metadata*.

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 based on 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

---

⋆ Corresponding author, email: `nkokash@liacs.nl`

demands on the users' familiarity and expertise in both semantic web technologies and the design principles of ontologies. It is unlikely that a user with *domain* expertise also has expertise in the *technologies* used to manage ontologies.

For example, 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 a particular gene to understand the cause of a skeletal malformation; and, (iv) a bio-engineer 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 journal paper) to models bearing well-defined data (e.g. from a clinical trial) or sets of mathematical equations (which might be used as input for a simulation tool).

Automating the discovery of relationships, in terms of physiological meaning, between these kinds of description is a key goal to the physiology community. Therefore, this community is investing considerable effort in building ontologies for the annotation and semantic management of such resources. For example, a number of reference ontologies have been created to represent gene products [3], chemical entities [4], cells [5] and gross anatomy [6]. Together, these ontologies consist of hundreds of thousands of terms, such that the volume of semantic metadata arising from resource annotation is considerable. Unfortunately, conventional technology for the management of ontologies and metadata is not usefully accessible to physiology experts, as they involve unfamiliar, abstract technicalities. Having to become technically proficient with such technology is a burden few physiology experts can bear without losing touch with their long term goals. Rather, domain experts should be able to manage data based on a familiar perspective, in which its meaning is made explicit in terms of the expert's own knowledge; a long standing challenge in knowledge engineering.

The present work builds on past efforts on RICORDO [1]. Those efforts have led to the design of backend software for the management of physiology metadata, including physiology models, genomics data, and radiology and histology images. This software comprises four notionally separable components: (a) expert knowledge, (b) a data repository, (c) a metadata repository and (d) an ontology knowledge base able to perform reasoning over ontology relationships. The ApiNATOMY effort has emerged to provide an intuitive graphical frontend to RICORDO. In this paper, we present a web-based ApiNATOMY environment, allowing physiology experts to navigate through circuitboard visualizations of body components, and their cardiovascular and neural connections, across different scales. Overlaid on these schematics are graphical renderings of organs,

neurons and gene products, as well as mathematical process models, in support of biomedical knowledge management use cases discussed in the next section.

The remainder of the paper is structured as follows: Section 2 gives an overview of the ontology-, metadata- and data-resources that we focused on for the ApiNATOMY prototype, and outline key use-case scenarios that motivate our work. Sections 3 and 4 then discuss the visualization techniques we applied to arrange and display those resources. Section 5 provides some insight into the implementation of the prototype. Finally, Sections 6 and 7 offer an overview of related efforts in the field, and conclude the paper with a discussion of the anticipated implications of our tool, as well as planned future work.

## 2   Use Cases and Data Resources

In this section, we briefly discuss a core use case for the ApiNATOMY application: the generation of interactive schematics in support of genomics and drug discovery studies. We introduce some of the key ontology- and data-resources required in this case. In so doing, we set the stage for an exposition of our early-stage results in the ApiNATOMY application effort.

The domains of genomics and drug discovery are heavily dependent on physiology knowledge, as both domains take into account the manufacture of proteins in different parts of the body and the transport of molecules that interact with those proteins, such as drugs, nutrients, and other proteins. We aim to provide an interactive, schematic overview of data resources important to these domains. This includes gene expression data (e.g. [7]), and data on the transport routes taken by molecular interactors (e.g. [8]). Such data may be usefully depicted in the form of a physiology *circuitboard*.

In ApiNATOMY, a physiology circuitboard schematic consists of an *anatomical treemap* and an overlay of *process graphs*. Our earlier prototypes [9,10] presented treemaps of the Foundational Model of Anatomy (FMA) ontology [6]. Nesting of one treemap tile inside another indicated that the term associated with the child tile is either a mereotopological *part* or a *subclass* of the term associated with the parent tile. Our newest prototype also adopts this convention.

The ApiNATOMY Graphical User Interface (Figure 1(a)) supports user interaction with circuitboard schematics via point-and-click navigation of the treemap content. The upper level of the anatomical treemap 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. The GUI supports data filtering across multiple levels and contextual zooming into selected areas.

This type of interaction extends also to the overlayed process graphs. These graphs project routes of blood flow processes linking different regions of the human body —using data generated in [11]—, as well as transport processes along neurons of the central nervous system (i.e., the brain and spinal cord) — using data obtained via the Neuroscience Information Framework [12].
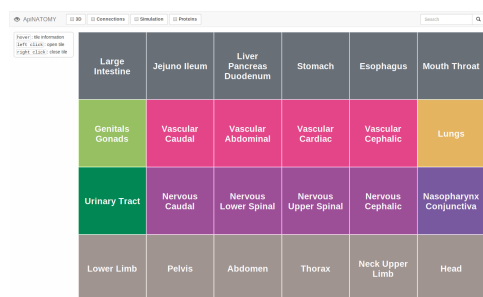
(a) Initial view of ApiNATOMY
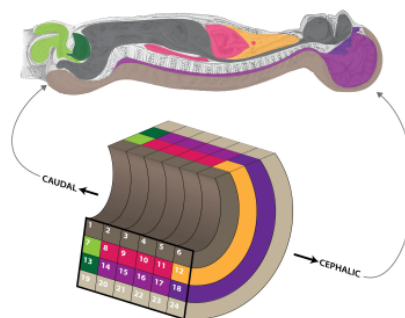
(b) Longitudinal section through the male human body, justifying the layout

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

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]. Easy access to such visual resources is critical to the understanding of long-range molecular processes in genomics and drug discovery research.

In the next two sections, we discuss our techniques for (i) constraining treemap layouts to generate stable anatomical treemaps, (ii) designing and overlaying physiological communication routes for the cardiovascular and neural systems, (iii) querying and depicting protein architecture diagrams for the anatomical overview of gene expression data, and (iiii) running simulations using mathematical models relevant to molecular mechanisms, and overlaying the results.

3   Visualizing Ontologies 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 resource, and consists of approximately 11,300 edges and over 10,000 distinct nodes. In this graph, 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 biological type of segment (1 - for arterial segment, 2 - microcirculation (MC), 3 - venous, and 4 - cardiac chamber), the third represents FMA IDs, the fourth and fifth are unique node identifiers in an edge pair, and the last column is a free-text description.

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

Table 1. Vascular connectivity data from the FMA ontology

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 (i.e., wall of left ventricle (FMA ID 7101), left atrium (7097), right ventricle (7098) or right atrium (7096)) 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 $7101 \rightarrow 2406 \rightarrow \cdots \rightarrow 8499 \rightarrow 62528$, and the path from there to the right atrium is $7096 \leftarrow 771 \leftarrow \cdots \leftarrow 9546 \leftarrow 9547 \leftarrow 62528$.

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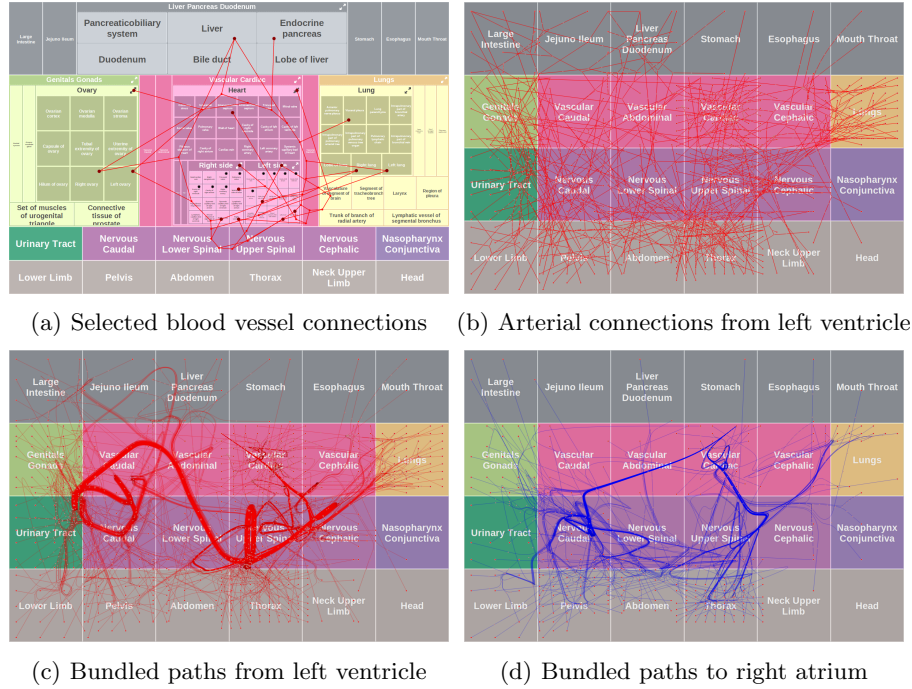
If there are too many edges to get a clear overview of the data —as in Figure 2(b), which shows the full connectivity graph for the left ventricle (7101) on the top-level body plan— we can apply a hierarchical edge bundling method that uses the path structure to bundle common sub-paths. The result for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

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

(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

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".

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 [22]. 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)).
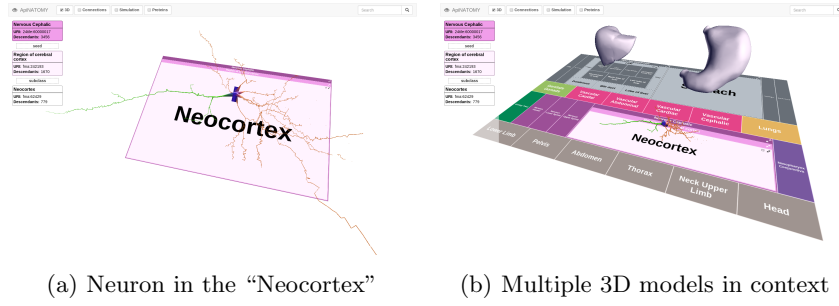
(a) Neuron in the "Neocortex"   (b) Multiple 3D models in context

Fig. 3. Visualization of static 3D models

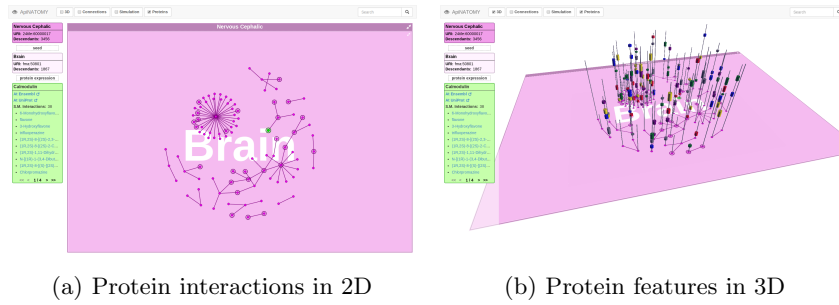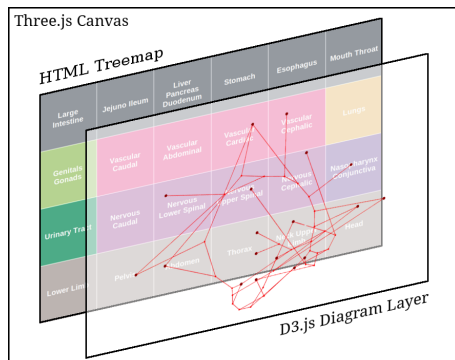(a) Protein interactions in 2D   (b) Protein features in 3D

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

Able to Control Position:

Three.js Canvas $\longrightarrow$ HTML Treemap

Three.js Canvas $\longrightarrow$ D3.js Layer

Able to Add Content:

HTML Treemap $\longrightarrow$ D3.js Layer

HTML Treemap $\longrightarrow$ Three.js Canvas

D3.js Layer $\longrightarrow$ Three.js Canvas

Fig. 5. The three layers of circuitboard visualization and their interaction.

5   Implementation

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.

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 in Figure 4(b). To render `.swc` neuron files (Figure 3), ApiNATOMY uses SharkViewer, an open source Three.js library developed by the Howard Hughes Medical Institute [23].

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 information 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 [24,25]. In addition, numerous anatomy-related taxonomies and databases have been created and are widely used by researchers in the biomedical field [26]. 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 [27] 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 [28] 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 [29] 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., [18,19,30]) 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

The core goal for ApiNATOMY is to put clinicians, pharmacologists, basic scientists and other biomedical experts in direct control of physiology knowledge management (e.g. in support of integrative goals outlined in [31]). As the domain of physiology deals with processes across multiple anatomical scales, the schematic ApiNATOMY approach provides a more flexible and customizable depiction of process participants, and the routes they undertake, compared to conventional methods of anatomy navigation that constrain visualization to regional views of very detailed and realistically proportioned 3D models (such as Google Body [32]). In this paper, we presented our initial results in the development of a generic tool that creates an interactive topological map of physiology communication routes. These routes are depicted in terms of (i) treemaps derived from standard reference anatomy ontologies, as well as (ii) networks of cardiovascular and neural connections that link tiles within these treemaps. These topological maps, also known as circuitboard schematics, set the stage for the visual management of complex genomic and drug-related data in terms of the location of gene products and the route taken by molecules that interact with them. While the implementation of our tool is still in its early stages, we have already started taking steps in preparation for future developments that will support:

– the visually-enhanced construction of mathematical models in systems biology (e.g. as discussed in [33]),
– the collaborative graphical authoring of routes of physiology communication (e.g. brain circuits) and, crucially,
– the automated calculation of relevant transport routes given (i) a fixed-location receptor and (ii) its corresponding ligand that is found elsewhere in the body.

Above all, our aim is to ensure that ApiNATOMY is easy to use by biomedical professionals, and available across a wide range of platforms, to foster collaborative exchange of knowledge both within, and between, communities of physiology expertise.

References

1. de Bono, B., Hoehndorf, R., Wimalaratne, S., Gkoutos, G., Grenon, P.: The RI-CORDO 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., Muthukr-ishnan, 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. Harnisch, L., Matthews, I., Chard, J., 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 knowl-edge 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** (2009) D782–D785
14. Ascoli, G.A.: Mobilizing the base of neuroscience data: the case of neuronal mor-phologies. Nat. Rev. Neurosci. **7**(4) (2006) 318–324
15. Johnson, B., Shneiderman, B.: Tree-maps: a space-filling approach to the visual-ization 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 multi-scale navigation techniques. TVCG **13** (2007) 1248–1253
17. Holten, D.: Hierarchical edge bundles: Visualization of adjacency relations in hier-archical 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. Soft-ware Practice and Experience **21**(11) (1991) 1129–1164

21. Bostock, M.: Sticky force layout. Online visualization tool (accessed on 20-May-2014)
22. EBI: Ensemble. Online web page (2014) accessed on 20-May-2014.
23. Weaver, C., Bruns, C., Helvensteijn, M.: SharkViewer. Howard Hughes Medical Institute, Janelia Farm Research Campus. DOI: 10.5281/zenodo.10053 (2014)
24. Magnenat-Thalmann, N., Ratib, O., Choi, H.F., eds.: 3D Multiscale Physiological Human. Springer (2014)
25. Magnenat-Thalmann, N., ed.: Modelling the Physiological Human. In Magnenat-Thalmann, N., ed.: Proceedings of the Second 3D Physiological Human Workshop. Volume 5903 of LNCS., Springer (2009)
26. Burger, A., Davidson, D., Baldock, R., eds.: Anatomy Ontologies for Bioinformatics. Volume 6 of Computational Biology. Springer (2008)
27. Burch, M., Diehl, S.: Trees in a treemap: Visualizing multiple hierarchies. In: Proc. VDA 2006. (2006)
28. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall (1999)
29. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. Comput. Graph. Forum **28**(3) (2009) 983–990
30. Selassie, D., Heller, B., Heer, J.: Divided edge bundling for directional network data. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis) (2011)
31. Hunter, P., et al.: A vision and strategy for the virtual physiological human in 2010 and beyond. Philos. Trans. A Math. Phys. Eng. Sci. **368**(1920) (2010) 2595–614
32. Wikipedia: Zygote body. Online web page (2014) accessed on 20-May-2014.
33. de Bono, B., Hunter, P.: Integrating knowledge representation and quantitative modelling in physiology. Biotechnol. J. **7**(8) (2012) 958–72