OXFORD

Data and text mining

VISUALGRAPHX: interactive graph visualization within Galaxy

Richard A. Schäfer and Björn Voß*

Institute of Biochemical Engineering, Computational Biology, University of Stuttgart, Allmandring, 31, 70569 Stuttgart, Germany

*To whom correspondence should be addressed.
Associate Editor: Jonathan Wren

Abstract

Motivation: We developed VISUALGRAPHX, a web-based, interactive visualization tool for large-scale graphs. Current graph visualization tools that follow the rich-internet paradigm lack an interactive and scalable visualization of graph-based data. VISUALGRAPHX aims to provide a universal graph visualization tool that empowers the users to efficiently explore the data for themselves at a large scale. It is available as a visualization plugin for the Galaxy platform, such that VISUALGRAPHX can be integrated into custom analysis pipelines.

Availability and Implementation: VISUALGRAPHX has been released as a visualization plugin for the Galaxy platform under AFL 3.0 and is available with instructions and application data at http://gitlab.com/comptrans/VisualGraphX/.

Contact: bjoern.voss@ibvt.uni-stuttgart.de

1 Introduction

Any biological system carries out its function through an elaborate interaction of a multitude of molecular components that together form complex biological networks. Visualization of these networks is of great interest in modern biology as it helps to gain insight into complex biological processes. Zhu *et al*. (2007) provide an overview of some of the major biological networks for which graphs are an appropriate mathematical representation (Pavlopoulos *et al*., 2011). As of today, various desktop-based applications that are able to visualize large-scale graphs exist. Most notable are CYTOSCAPE (Shannon *et al*., 2003), GEPHI (Bastian *et al*., 2009) and TULIP (Auber *et al*., 2012). In addition, the GRAPHVIZ software package provides a comprehensive collection of open-source tools for the visualization of graphs on the basis of the DOT graph description language (Ellson *et al*., 2004). It is widely distributed, because it can be called from general purpose languages via specific interfaces. However, the emergence of the internet as the default software platform triggered a shift toward Rich Internet Applications. These provide a rich and interactive user experience in a cross-platform manner through a standard web browser. Visualization of large-scale graphs benefits from this paradigm as web development technologies facilitate interactivity and scalability. TOUCHGRAPH® (Touchgraph, LLC, USA), TOM SAWYER VISUALIZATION® (Tom Sawyer Software, USA) and MANY EYES (Viegas *et al*., 2007) follow the rich-internet paradigm and are able to visualize graph data in a sophisticated manner, but still lagging behind the possibilities offered by current web standards and are, beyond that, not optimized for biological networks. Similarly, Cytoscape Web (Lopes *et al*., 2010) is a web-based network visualization tool that is modeled after Cytoscape. HYPERSCAPE (Cromar *et al*., 2015) implements hypergraphs to capture complex hierarchical structures, but has its limitations in the generalization for biological networks. Nevertheless, hypergraphs provide an interesting concept both in the visualization and analysis of biological networks (Klamt *et al*., 2009). The Galaxy system for biological data analysis (Giardine *et al*., 2005), for a long time, did not offer graph visualization. Recently, an effort for the visualization of graphs (GraphVis, https://github.com/eteriSokhoyan/GraphVis) has been integrated into the Galaxy Main instance. It is based on the cytoscape library for JavaScript (Franz *et al*., 2016). We aim to contribute to this field and present VISUALGRAPHX for the interactive exploration of large-scale graphs in a scalable manner that utilizes current web standards to provide a rich user interaction.

2 Implementation

VISUALGRAPHX follows the Model-View-Controller (MVC) concept in order to isolate the data (M) from the presentation (V) and its manipulation (C). Gamma *et al.* (1994) consider MVC as an architectural pattern that merely expresses the separation of concerns, while allowing it to be adapted to specific needs. It has been widely adopted as an architecture for rich-internet applications due to a variety of open-source frameworks that make use of this pattern. Although the interpretation of MVC distinguishes these frameworks from one another, the basic concept stays the same. VISUALGRAPHX utilizes the Backbone.js library for JavaScript that provides an unopinionated set of primitives to build single-page applications.

The JSON graph format (http://jsongraphformat.info) is an appropriate file format for describing graphs where the nodes and edges are deposited as arrays of objects. In addition, its usage eases the processing of the data as Backbone.js provides a RESTful JSON interface. Visualization of graphs requires algorithms to determine the arrangements of its components in the multi-dimensional space. Multiple layout strategies are used in practice, but the force-directed layout (Kobourov, 2012) seems to be the most prominent one. In VISUALGRAPHX we use the D3 javascript (http://www.d3js.org) library, which implements the force-directed layout with a high degree of freedom using position Verlet integration. However, D3 does not offer graph traversal algorithms, which we implemented primarily as depth-first and breath-first searches. Shneiderman (1996) coined the information seeking mantra "Overview first, zoom and filter, then details-on-demand", that we fulfill in both ways. In more detail, upon setting the parameters for graph depth and the start (root) node, VISUALGRAPHX starts to precompute the corresponding subgraph, which is displayed upon clicking "visualize". Here, the precomputed subgraph can be expanded, contracted and moved by common user interactions, such as double-clicks on nodes. As opposed to other graph visualization environments this decoupling ensures a high efficient buildup of the graph, thus providing a high degree of scalability. VISUALGRAPHX achieves a high degree of flexibility as it allows nodes to be embedded as single predefined scalable vector graphics or as arbitrary external graphics when specified as Uniform Resource Locator (URL) in the nodes attributes of the JSON data. Furthermore, asynchronous method dispatch enables VISUALGRAPHX to update the subgraph without delay.

3 Results

We used VISUALGRAPHX for the visualization of the results of COVENNTREE (Lott et al., 2015), which is a method for the comparative analysis of large datasets (Fig. 1). COVENNTREE generates a rooted tree based on the NCBI taxonomy, in which the nodes are associated with weighted Venn diagrams to illustrate the relation of the different datasets. Creation of the diagrams is outsourced, because they are given as URLs to the Google Chart API, which VISUALGRAPHX can handle directly. The output of COVENNTREE consists of a network file that defines the structure of the tree and a corresponding attribute file that contains the properties of the nodes. We wrote a tool that converts these files to the JSON graph format, namely vgx_converter installable through the Galaxy Toolshed. As VISUALGRAPHX precomputes an initial subgraph as opposed to loading the full graph at once it accelerates the buildup of large-scale graphs. Loading a graph that contains 4692 nodes and 7688 edges (mTOR_ChEMBL.jgf) with VISUALGRAPHX took 6 s, while loading into CYTOSCAPE took roughly 15 s, which corresponds to 2.5× speedup. We further increased the network size to 100K nodes/egdes
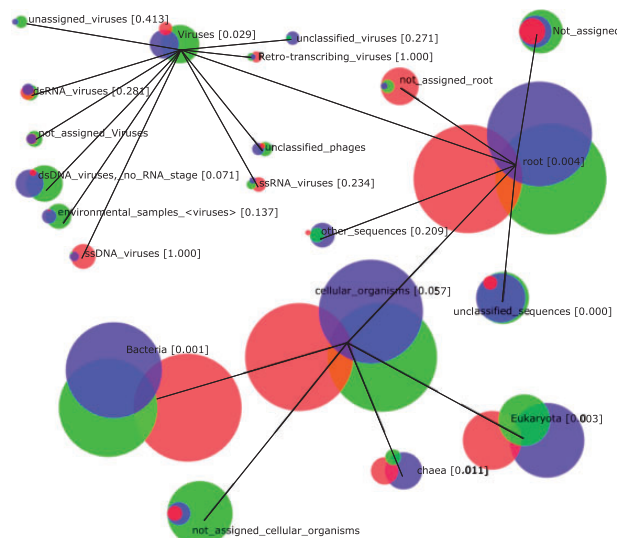

Fig. 1. Snapshot of a VISUALGRAPHX visualization. We took the results from a comparative analysis of three 16S ribosomal RNA datasets (Steglich et al., 2015) with CoVennTree (Lott et al., 2015) and classified according to the NCBI taxonomy. Weighted Venn diagrams at the nodes represent the relations between the three datasets, taxon names are given and augmented by the so called Venn Decomposition Score (see Lott et al., 2015 for details) in square brackets

for which the buildup of the graph took about 180 s. For further illustration of the general applicability of VISUALGRAPHX we visualized data from the CAUSAL BIOLOGICAL NETWORKS DATABASE (Boué et al., 2015). We have converted several different biological networks from this database and deposited them in the repository for visualization with VISUALGRAPHX.

4 Conclusion

With VisualGraphX we present a general visualization plugin for the Galaxy platform through which it can be easily integrated in any bioinformatics pipeline where interactive graph visualization is of interest. VISUALGRAPHX has been structured with regard to the MVC concept and utilizes internal libraries of Galaxy to accurately fit to the platform. For the visualization we made use of widespread web standards that allow us to provide a more dynamic and interactive visualization of large-scale graphs than current tools. Furthermore, the structure of VISUALGRAPHX allows it to be extended for more specific use cases of graph visualization and its streaming design, where data is loaded and visualized on-demand, ensures its performance even for large scale graphs.

References

Auber,D. *et al.* (2012) The Tulip 3 framework: a scalable software library for information visualization applications based on relational data. Research report RR-7860, INRIA, p.31.

Bastian,M. *et al*. (2009) Gephi: an open source software for exploring and manipulating networks. In *3th International AAAI Conference on Web and Social Media*, AAAI Press.

Boué,S. *et al*. (2015) Causal biological network database: a comprehensive platform of causal biological network models focused on the pulmonary and vascular systems. *Database*, **2015**, bav030.

Cromar,G.L. *et al*. (2015) Hyperscape: visualization for complex biological networks. *Bioinformatics*, **31**, 3390–3391.

Ellson,J. *et al*. (2004) Graphviz and dynagraph—static and dynamic graph drawing tools. In Jünger, M. *et al*. (eds.), *Graph Drawing Software, Mathematics and Visualization*, Springer, Berlin, pp. 127–148.

Franz,M. *et al*. (2016) Cytoscape.js: a graph theory library for visualisation and analysis. *Bioinformatics*, **32**, 309–311.

Gamma,E. *et al*. (1994) *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st edn. Addison-Wesley Professional, Reading.

Giardine,B. *et al*. (2005) Galaxy: a platform for interactive large-scale genome analysis. *Genome Res*., **15**, 1451–1455.

Klamt,S. *et al*. (2009) Hypergraphs and cellular networks. *PLoS Comput. Biol.*, **5**, e1000385.

Kobourov,S.G. (2012) Spring Embedders and Force Directed Graph Drawing Algorithms. ArXiv12013011 Cs.

Lopes,C.T. *et al*. (2010) Cytoscape web: an interactive web-based network browser. *Bioinformatics*, **26**, 2347–2348.

Lott,S.C. *et al*. (2015) CoVennTree: a new method for the comparative analysis of large datasets. *Front. Genet*., **6**, 43.

Pavlopoulos,G.A. *et al*. (2011) Using graph theory to analyze biological networks. *BioData Min*., **4**, 10.

Shannon,P. *et al*. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*., **13**, 2498–2504.

Shneiderman,B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, IEEE Comput. Soc. Press, Silver Spring, pp. 336–343.

Steglich,C. *et al*. (2015) Dataset for metatranscriptome analysis of Prochlorococcus-rich marine picoplankton communities in the Gulf of Aqaba, Red Sea. *Mar. Genomics*, **19**, 5–7.

Viegas,F. *et al*. (2007) ManyEyes: a site for visualization at internet scale. *IEEE Trans. Vis. Comput. Graph*., **13**, 1121–1128.

Zhu,X. *et al*. (2007) Getting connected: analysis and principles of biological networks. *Genes Dev*., **21**, 1010–1024.