Applying Heat Maps in a Web-Based Collaborative Graph Visualization

Björn Zimmer* Andreas Kerren†

ISOVIS Group, Department of Computer Science, Linnaeus University, Växjö, Sweden

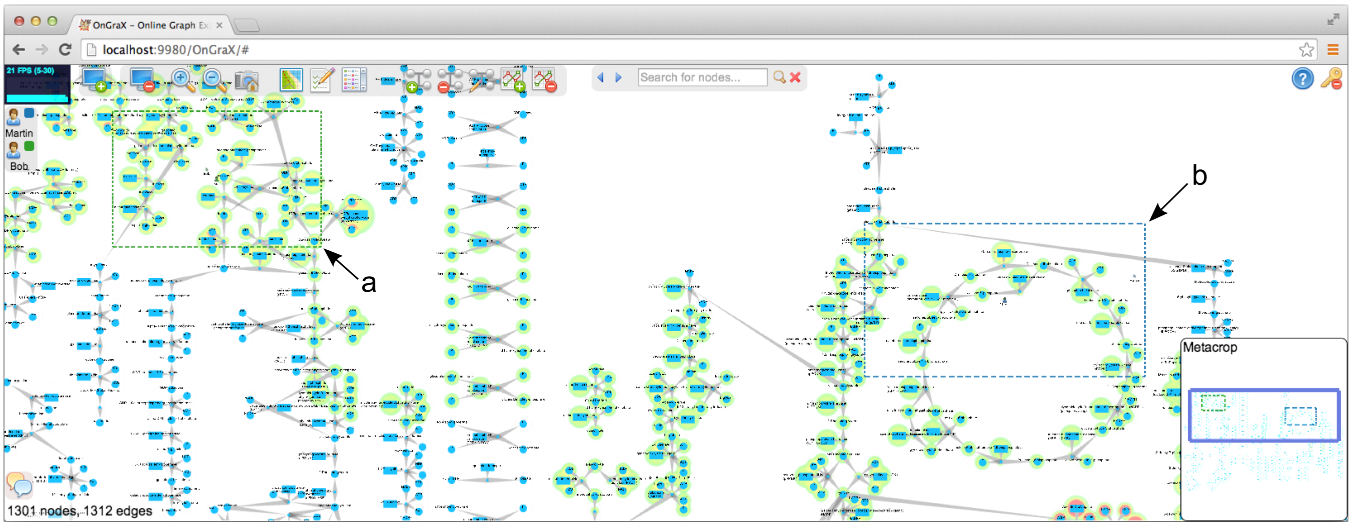

Figure 1: Screenshot of a biochemical network visualized by our collaborative graph visualization system OnGraX. The green (a) and blue (b) dashed rectangles represent viewing areas of two other users ('Bob' and 'Martin') who are exploring this graph simultaneously. In this concrete case, the underlying heat map highlights those nodes that have been of interest to all users during the last hour. The overview in the bottom right corner shows the entire input graph.

Abstract

The visual analysis of large and complex networks is a challenging task in many fields, such as systems biology or social sciences. Often, various domain experts work together to improve the analysis time or the quality of the analysis results. Collaborative visualization tools can facilitate this process. We propose a new web-based visualization environment which supports distributed, synchronous and asynchronous collaboration for graphs with up to 10,000 nodes and edges. In addition to standard collaboration features like event tracking or synchronizing, our client/server-based system provides visualization and interaction techniques for better navigation, guidance and overview of the overall data set. During asynchronous collaborations, network changes made by specific analysts or even just visited elements are highlighted on demand by heat maps. These heat map representations are user-sensitive in a sense that the current analyst is able to perceive which changes were made by others.

Keywords: Network exploration, graph drawing, interaction, heat maps, collaboration, WebGL, CSCW

1 Introduction

With the growing size and availability of large and complex data, the cooperative analysis of such data sets is becoming increasingly important for many data analysts. Those people—who are often spread across the globe—work together in order to analyze data sets efficiently. The analysis process can either take place in a joint online session where everybody is working simultaneously on one data set, discussing and changing it together in real-time. Or the analysts work on the data set whenever they find the time to avoid having to schedule and organize a virtual or physical meeting with a larger group of colleagues. Both situations cause specific problems that should be handled by tools that support collaborative work. For instance, while working independently, it would be helpful to see changes of the data performed by other analysts. Another interesting issue is to see which part of the data set was already explored by others. Here, it is also interesting to know who has changed the data: was it an established expert working on a specific part of the data, or a new staff member who might not have the same experience as the expert? On the other hand, in a scenario where analysts work simultaneously together, the different experts want to see what the others are doing and if there are possibilities to coordinate the efforts in some way to get better analysis results.

Providing a suitable environment for those collaborative settings is an interesting challenge [7]. Most existing visualization systems are usually designed for single users and only some systems feature extensions for collaborative work, such as [2, 4]. These tools often have to be installed on every machine, run in Java applets embedded in web pages, or require external browser plugins. Collaborative visualization tools should instead be easy to setup on the fly without considerable overhead. This problem was mentioned by Isenberg et al. [3] as one of the ongoing challenges in collaborative visualization. Fortunately, recent developments in web-based technologies offer new and powerful approaches for running interactive applications directly in a browser without the need to install any additional software. With such an application at hand, it suffices to send an URL to the collaboration partners to start a collaborative session which provides a very easy way to perform a collaboration.

*e-mail: bjorn.zimmer@lnu.se
†e-mail: andreas.kerren@lnu.se

Poster Abstract, IEEE Information Visualization (InfoVis '14), Paris, France, 2014.

Our work is based on a client-server architecture, see Fig. 2. Clients are responsible for rendering the visualization in WebGL, providing the user interface, and sending action events and the user's viewing area to the server which handles the data storage and distributes all events among the connected clients. The rendering of the node-link diagrams is achieved with three.js [6] and scales up to 10,000 nodes and edges.

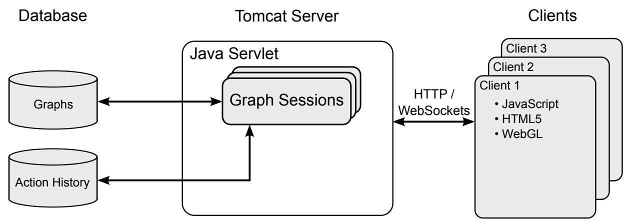

Figure 2: General architecture of our system. Sessions are initialized on demand whenever a user joins a graph analysis session. The complete action history of a graph is stored in a database. Client-server communication via WebSockets ensures the real-time propagation of changes done to a graph across all connected clients.

2 Collaborative Visualization

Synchronous sessions in our tool are based on the work of Gutwin and Greenberg [1]. Users are able to see each other's views and mouse positions and can also automatically move the camera to another user's view. If a user marks a node or changes something in the graph, this will be indicated to all other users with short animations and notification icons in the right screen corner. For instance, deleted nodes will slowly fade out instead of just disappearing. By clicking on a notification icon, the user can quickly move the camera back and forth between his/her current working location and the position of the respective event. While working *asynchronously*, users can also leave annotations in the graph area or directly at nodes (see Fig. 3 (a)) to propagate their thoughts and findings about specific parts of the data to subsequent analysts. This also works for chat messages which can be bound to a position or node in the graph.

2.1 Visualizing user behavior data with heat maps

We use a heat map-based visualization to enable analysts the exploration of all logged events of a graph analysis session. The heat map can be computed by using two different options. The first option calculates a heat map based on the amount of seconds nodes have been viewed by users. In this case, all logged user views and node positions are correlated to calculate the heat map values, thus making it robust against changes in the layout of the graph. For zoomed-out views that show a lot of nodes, it is clear that the user does not attend to all nodes in such a view. We solve this issue by taking only those views into account that lie below a user-specified threshold. This approach does still include nodes in the zoomed-in views that had no attention by the user, but it gives a rough estimate about the viewed graph regions without asking a user to mark every inspected node manually or asking all users to use an eye tracker during the analysis process, for instance. The second option calculates a heat map based on manual changes performed on nodes in a graph (e.g., node movement, name changes, node selections, or adding/removing edges). A multiplier can be specified for each individual event type to give it more or less weight during the calculation.

Fig. 3 shows an example of the heat map visualization. It can be configured to only show the data of particular users. Furthermore, it is possible to select a time frame, for instance, the last hour of the current analysis session, or a specific start and end date. This enables an analyst to review changes done in a collaborative session during a particular time frame or to check the work of specific users.

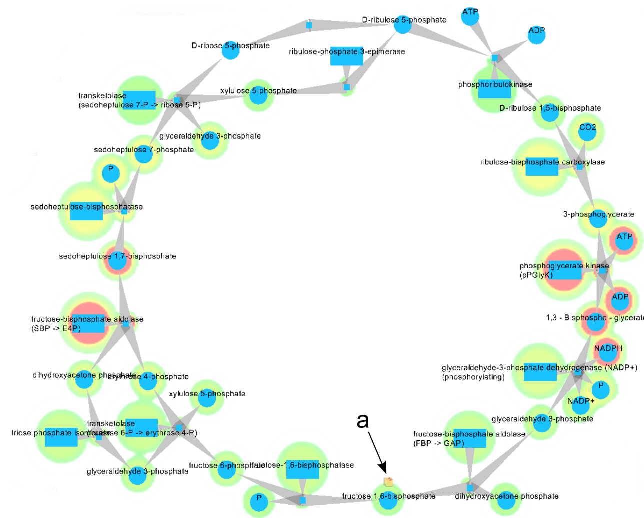

Figure 3: A heat map is used to indicate which parts of a graph were viewed or changed by other users. Here, the nodes on the top were not much visited/changed. The arrow (a) highlights a marker pinned to a node. It can be clicked to open a textual annotation that was added to this node by a former analyst.

3 Conclusion

In this work, we presented a collaborative system for visualizing graphs in a web-based environment. By using a client/server architecture, users do not have to install any additional applications or browser plugins. The start of a collaborative analysis session is simply done by opening a URL in a browser window. With the help of our tool, users can seamlessly join or leave graph sessions on the fly and do not have to wait for other users to start or finish their work. All actions performed during a session as well as the users' camera positions are tracked and can be visualized along with the graph data by using heat map representations which are robust against changes in the graph layout. To provide data-aware annotations [5], textual markers and chat messages can be linked to nodes or arbitrary positions, helping experts to analyze regions of a graph that were of interest or have been edited by former analysts.

References

[1] C. Gutwin and S. Greenberg. Design for individuals, design for groups: Tradeoffs between power and workspace awareness. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW '98, pages 207–216, New York, NY, USA, 1998. ACM.

[2] P. Isenberg, A. Bezerianos, N. Henry, M. S. J. Carpendale, and J.-D. Fekete. CoCoNutTrix: Collaborative Retrofitting for Information Visualization. *IEEE Computer Graphics and Applications*, 29(5):44–57, 2009.

[3] P. Isenberg, N. Elmqvist, D. Cernea, J. Scholtz, K.-L. Ma, and H. Hagen. Collaborative Visualization: Definition, Challenges, and Research Agenda. *Information Visualization*, 10(4):310–326, 2011.

[4] S. Jourdain, U. Ayachit, and B. Geveci. ParaViewWeb, A Web Framework for 3D Visualization and Data Processing. *IADIS International Conference on Web Virtual Reality and Three-Dimensional Worlds*, 3:870–877, 07 2010.

[5] A. Kerren and F. Schreiber. Toward the role of interaction in visual analytics. In *Proceedings of the Winter Simulation Conference*, WSC '12, pages 420:1–420:13. Winter Simulation Conference, 2012.

[6] Ricardo Cabello. `three.js`. Last accessed June 22, 2014.

[7] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, August 2005.