Displaying User Behavior in the Collaborative Graph Visualization System OnGraX

Björn Zimmer and Andreas Kerren

Department of Computer Science, ISOVIS Group, Linnaeus University,
Vejdes Plats 7, SE-35195 Växjö, Sweden
{bjorn.zimmer,andreas.kerren}@lnu.se

Abstract. The visual analysis of 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 the analysis process in such situations. We propose a new web-based visualization environment which supports distributed, synchronous and asynchronous collaboration. In addition to standard collaboration features like event tracking or synchronizing, our client/server-based system provides a rich set of visualization and interaction techniques for better navigation and overview of the input network. Changes made by specific analysts or even just visited network elements are highlighted on demand by heat maps. They enable us to visualize user behavior data without affecting the original graph visualization. We evaluate the usability of the heat map approach against two alternatives in a user experiment.

Keywords: Information visualization, graph drawing, network exploration, interaction, HCI, CSCW, biological networks, heat maps

1 Introduction

With the growing size and availability of large and complex data, the cooperative analysis of such data sets is becoming an important new method for many data analysts as cooperation might improve the quality of the analysis process [15] and help to analyze data sets efficiently. One crucial observation is that collaborators—who are often spread across the globe—would like to seamlessly drop in and out of ongoing work [13]. On the one hand, the collaborative analysis process can take place in a joint online session where everybody is working simultaneously on one data set, discussing and changing it together in real-time to create better analysis results. Here, different experts might want to see what the others are doing, and if there are possibilities to coordinate their efforts and find a common ground [3,9]. On the other hand, the experts work on the data set whenever they find the time (i.e., asynchronously) 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 which

In Proceedings of the 23th International Symposium on Graph Drawing & Network Visualization (GD '15), pages 247-259, volume 9411 of LNCS, Los Angeles, CA, USA, Springer, 2015.

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 has already been explored by others. Here, it is also interesting to know who changed the data: was 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?

To tackle the aforementioned problems in the context of collaborative network analyses, we have developed the visualization tool OnGraX [23,24,25]. Our system was designed for the *distributed asynchronous* and *synchronous* collaborative exploration of graphs in a modern web browser. Note that we give a detailed explanation about the engineering aspects of OnGraX in paper [25]. In contrast, we here propose interactive visualization techniques that

- help to coordinate work in a collaborative setting for *node-link diagrams* which may change their topology during the analysis process (referred to as dynamic graphs in the following) and
- assist analysts to identify previous activities performed by former users on these networks.

We exemplify our visualization approaches with the help of the collaborative analysis of metabolic networks from the Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway database [1] due to our long lasting research collaborations with biologists/bioinformaticians at several research institutions. Building such biological networks is often based on complex experiments. In consequence, biologists of different domains and experience levels want to explore the resulting networks and check them for wrong entries or missing data and revise the networks wherever it is necessary. Usually, they only check parts of a network that are specific to their own field of expertise or interest. In this case it is important to know, what part of the network has already been checked and what part still needs attention. This can also be used as a kind of quality check: an area which has been investigated by many different experts is likely to have a higher quality than an area only investigated by one scientist. OnGraX supports such analysis tasks by providing methods for data awareness and coordination. Note that we retain this usage scenario in the rest of the paper except in the heat map evaluation (cf. Sect. 5) in order to attract a higher number of test subjects.

The remainder of this paper is organized as follows. In the next section, we discuss related work in collaborative graph visualization. We describe our design decisions in Section 3 and explain OnGraX' interaction and visualization techniques for displaying user behavior in Section 4. A user experiment to evaluate our heat map approach for identifying previous user activities is discussed in Section 5, and we conclude our paper in Section 6.

2 Related Work

Isenberg et al. [11] give a good overview of definitions, tasks and sample visualizations in the field of collaborative visualization. The authors define collaborative visualization as “the shared use of computer-supported, (interactive,) visual representations of data by more than one person with the common goal of contribution to joint information processing activities”. They also provide an excellent summary of ongoing challenges in this field. All discussed standard systems incl. more recent developments (e.g., ManyEyes [20] or Dashiki [17]) are not suitable for our collaborative analysis problems, since they do not support the interactive visualization of node-link diagrams in a web browser with real-time interactions for collaboration. The benefits of collaborative work were also discussed in an article on social navigation presented by Dieberger et al. [6]. Being able to see the usage history and annotations of former users might help analysts to filter and find relevant information more quickly. In order to be able to work together during a synchronous session, users have to know each other’s interactions and views on the data set, usually referred to as “common ground” [3,9]. To find a common ground in node-link visualizations, we apply the techniques from the work of Gutwin and Greenberg [8]. They used secondary viewports and radar views to indicate other users’ view areas and mouse cursor positions. We use a similar approach and show the viewports of other users as rectangles in the background of the graph visualization. Another work by Isenberg et al. [12] introduced the concept of collaborative brushing and linking, which ”allows users to communicate implicitly, by sharing activities and progress between visualizations”. The authors considered sharing activities during synchronous collaborations on a tabletop visualization for document collections. We adapt the concept and utilize it in node-link diagrams with the help of a heat map visualization for the exploration of interaction information in both asynchronous and synchronous, distributed sessions.

Our tool OnGraX utilizes heat maps to analyze and identify highly frequented or edited parts of the graph based on user behavior. Patina [16] uses a similar approach but focuses on visualizing the usage of user interfaces, whereas our tool facilitates heat maps to visualize interactions of users with the data itself. To the best of our knowledge, heat map visualizations for representing data in combination with node-link diagrams are seldomly considered. Usually, they are used to visualize quantitative data in geovisualizations [19], as cluster heat maps [7], or for the visualization of eye tracking data to illustrate the quality of web site designs, user interfaces, or graph layouts [21,18], i.e., for evaluation purposes. One of the few examples where heat maps are used in node-link diagrams is PLATO [22] which employs heat maps to visualize gameplay data.

3 Design Decisions

We carefully designed our system in terms of visual representations, interaction techniques, and analysis processes to support biologists/bioinformaticians in exploring and curating graphs from the KEGG pathway database. We decided to focus our work on node-link diagrams, since this is still the most accepted and preferred graph drawing metaphor, and our users are familiar with this kind of visualization. Our overall goal was to develop a visualization system that allows analysts spatially spread across multiple research labs or even countries to quickly start an analysis session and to work on large and complex networks together. A special problem that arises during the distributed analysis of graphs is that topology and structure of a graph are independent to the layout. Analysts might change the layout drastically during the analysis process, which complicates the task of keeping track of the graph objects and areas that users were most interested in. We also want our tool to support tracking and subsequent visualizing of all actions and graph changes performed by the users. This includes to keep track of the users' camera positions and use this data later to assist users in finding parts of a graph that were interesting to other analysts or have been edited a lot. The reason behind this is that users in a collaborative working environment do not always find the time to work together simultaneously. They would prefer to work on the data set whenever it is convenient for them. And in such a case, they would like to review changes that have been performed on the data set by other analysts in the past. Maybe, they also want to find out which part of the data set another analyst was looking at, since he/she might be an expert in the underlying application field and has another exploration pattern compared to less experienced users. Showing this data—the camera and mouse positions, the logged user views, and changes to specific objects—in the graph without changing the original node-link visualization was an important requirement for our users. Biologists are accustomed to existing layouts and drawing conventions of graphs from the KEGG pathway database. Thus, changing positions, color, or the shape of nodes to show the data which is collected during collaborations is not an option for our analysis tasks.

During their work, analysts would also like to share their thoughts, insights, and questions about specific nodes, edges or regions with other users. This could happen during a *synchronous* session where collaborators want to discuss their findings, or in an *asynchronous* session where users would like to share messages and pointers on specific nodes. Heer and Agrawala discuss these ideas as "Common Ground and Awareness" and "Reference and Deixis" in their work on collaborative visual analytics [9]. In case of graphs that change their topology during the analysis process, single nodes or complete graph regions could be deleted from a graph, rendering old user annotations useless without the possibility to view them in their historical context. Thus, analysts need a way to quickly view the graph in a state when the annotation was originally written. Based on this discussion, we categorize our requirements as described in the following.

Collaboration Requirements (C-R)

1. Users should be aware of the position of other users in the same synchronous session.
2. Users should have possibilities to establish and keep a common ground with other users. Everyone should be aware of performed changes on the graph during a session.
3. They should have an option to discuss ongoing work through persistent chat channels and annotations.

Visualization Requirements (V-R)

1. Annotations should be viewable in their historical context. Thus, it should be possible for users to review old graph states.
2. Provide an easy and intuitive way for analysts to find out which regions of a graph where viewed and/or changed by former users.
3. Additionally, the visualization of this data should not interfere with the original node-link diagram.

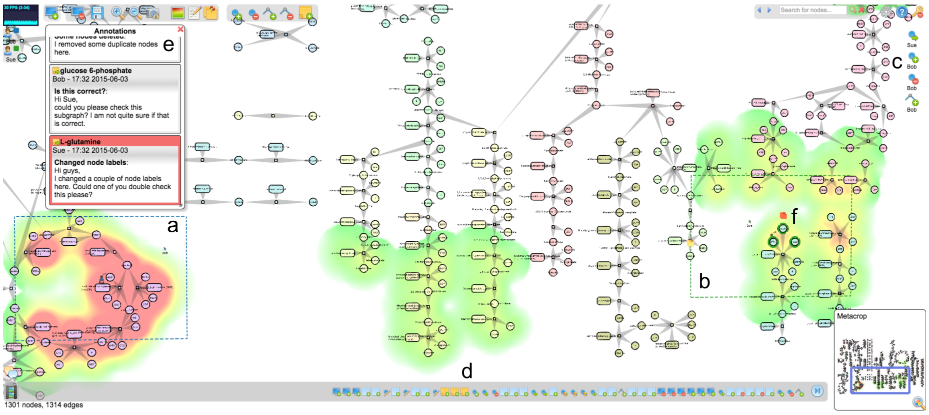
4 Interaction and Visualization Techniques

Figure 1 shows an overview of the tool right after joining an ongoing graph analysis session. In this case, the user has joined a session where two other users, Bob and Sue, are already working in. Their viewports are represented as two dashed rectangles: Bob's view is shown in blue (bottom left) and Sue's view is shown in green (bottom right). All users in a session are listed as small icons at the left hand side of the screen. By clicking on one of the user icons, the camera moves to his/her current position in the graph. This feature provides a quick way to join and discuss another user's viewing area. Visualizing the viewports of other users helps us to tackle our first collaboration requirement (cf. C-R 1). An overview of the graph is rendered in the bottom-right corner of the screen. Here, the user's camera position is shown as a blue rectangle. As in many other standard visualizations that use overview+detail [4], this rectangle can be dragged to another position in the overview in order to modify the detail view (the same can be done by simply clicking on the new position in the overview).

We use a standard node-link metaphor to visualize graphs in our system. The visualization uses tapered edges for directed graphs, as suggested by Holten and van Wijk [10], since they provide users with a faster way to find connected nodes as opposed to arrowhead edges. If another user selects one or more nodes, this will be visible to all other participants of the analysis session. An outline in the respective user color is added to a selected node; thereby the system adapts the outline shape to the corresponding node shape. To make graph changes performed by other users during a synchronous session more obvious and to address the second collaboration requirement (cf. C-R 2), we use short animations on the affected objects, similarly to the work of Gutwin and Greenberg [8]. For instance, the outlines for other users' node selections are animated shortly while they are added or removed, nodes are slowly moved to new positions instead of just jumping there after being moved by another user, and deleted nodes slowly vanish instead of just disappearing.

4.1 Annotations and Chat Links

In order to improve the communication among collaborators, our tool has a persistent chat channel for every graph session and offers the possibility to link chat messages to a position or a node in the graph. Users can use those chat



Some nodes deleted:
I removed some duplicate nodes here.

glucose 6-phosphate
Bob - 17:32 2015-06-03

Is this correct?:
Hi Sue,
could you please check this subgraph? I am not quite sure if that is correct.

L-glutamine
Sue - 17:32 2015-06-03

Changed node labels:
Hi guys,
I changed a couple of node labels here. Could one of you double check this please?

Fig. 1. Overview of our system. The image shows a part of a biochemical network with 1,301 nodes and 1,314 edges. The blue and green dashed rectangles (see (a) and (b)) are the viewing areas (viewports) of two other users who are also exploring this graph simultaneously. In this concrete case, the underlying heat map highlights those nodes that were in the viewing area to all users during the last hour. Symbols in the top-right corner of the screen (c) assist analysts to keep track of recent actions performed by other users. the timeline (d) is used to temporarily revert the graph to a previous state and to replay applied changes. Analysts can pin text annotations to nodes and edges to discuss tasks, insights and questions with each other (e+f).

links to move the camera to the linked object or position. A link to an arbitrary position might become obsolete after changes to the graph layout, but a message linked to a node or edge will always be valid as long as the object is not deleted. In addition, users can attach textual annotations directly to nodes or edges (cf. Figure 1, (e+f)). These annotations work as pointers from the graph visualization to text and vice versa. Clicking on an annotation in the graph visualization opens the annotation dialog and highlights the linked message. A click on an annotation in the dialog moves the camera to the object's position in the graph visualization. With the chat and annotation features, we address our last collaboration requirement (cf. C-R 3).

One problem with textual annotations and chat messages linked to objects is, that the original context in which an annotation or message was initially written could get lost if the respective graph region—where the link is pointing to—is changed during the course of a session or if the object with this link is deleted. We solve this problem by enabling analysts to temporarily revert the complete graph to an old state (similar to the timeline feature, cf. Sect. 4.3) by right clicking on a chat link or an annotation, giving them the possibility to view the graph in a state in which the annotation was originally written. This feature addresses our first visualization requirement (cf. V-R 1).

4.2 Visualizing User Behavior Data with Heat Maps

In order to provide users with a way to quickly find out which nodes or regions of a graph were viewed and/or changed by others (cf. V-R 2), we considered several options. It would be possible to map the corresponding data to the colors or the size of the nodes. Another option would be to use additional glyphs on/around the nodes which represent this data. Using glyphs would also allow us to show both the viewport data and the data for graph changes at the same time, as small bar charts for instance. The third option is a heat map-based visualization in the background of the graph visualization. We decided to omit mapping the data to the size of nodes, as this would interfere too much with the original graph layout and could introduce too many node overlaps. Additional options would have been to use contour lines [2] or bubble sets [5], but for our use case the focus usually lies on finding and marking single nodes instead of bigger regions in a graph. The remaining three options are exemplified in Figure 2.

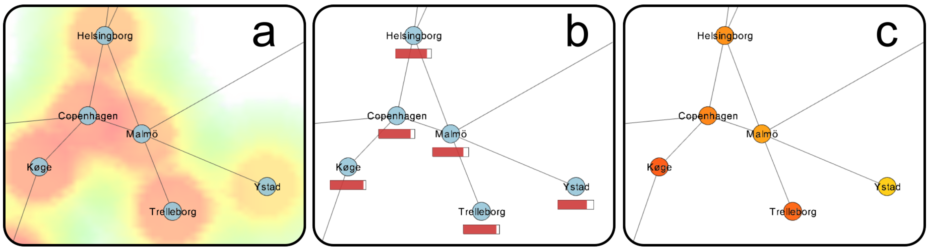


Fig. 2. Heat map visualization (a) and two alternative approaches: glyphs (b) and node color (c). They are used to indicate which parts of the entire graph were viewed or changed by other users.

One disadvantage of glyphs in this context is the increased clutter in the graph visualization. Additionally, depending on the size of the glyphs, it could be hard to see the actual data values in highly zoomed-out views of the graph. Changing the color coding of nodes in a graph as alternative is in conflict with our last visualization requirement (cf. V-R 3), because the color coding can be already mapped to another attribute. Thus, heat maps could provide a good alternative to visualize additional data without directly changing the attributes of objects in a node-link diagram. Users can choose between a colored heat map and a monochrome heat map in case the colored version interferes too much with the actual node colors. We performed a user experiment (cf. Sect. 5) to assess how the heat map approach compares against glyphs and node colors. The actual values, which are mapped to the glyphs, node colors, or heat map can be computed based on two different data sources: viewports and graph changes.

Displaying Viewports In the first case, values are calculated based on the amount of seconds that nodes have been in the viewing areas of users (visitation rate).

For aggregating this data, OnGraX stores each user's viewport together with the time spent on the position whenever the viewport is changed. Additionally, each time a node is moved, the old position is logged. The server correlates all logged user views and node positions to calculate the values, thus making them robust against changes in the layout of the graph. Figure 3 illustrates this approach. In this small example, three stored viewports of one user and two node movements from another user—whose viewports are ignored here—are taken into account. The user arrived at position A at exactly 10:00 AM, stayed there for 10 seconds, moved his viewport to position B for 5 seconds and finally stayed 16 seconds at position C. In viewport A, node 1 was visible for 10 seconds, but in viewport C, it was only visible for 12 seconds, as the node was only moved into the viewport 4 seconds after the user arrived at the position, resulting in a complete viewing time of 22 seconds for node 1. The viewing time of node 2 is only 2 seconds, as it was moved into viewport B 13 seconds after 10:00 AM, and the user arrived there at 10 seconds after 10:00 AM and left 5 seconds later.

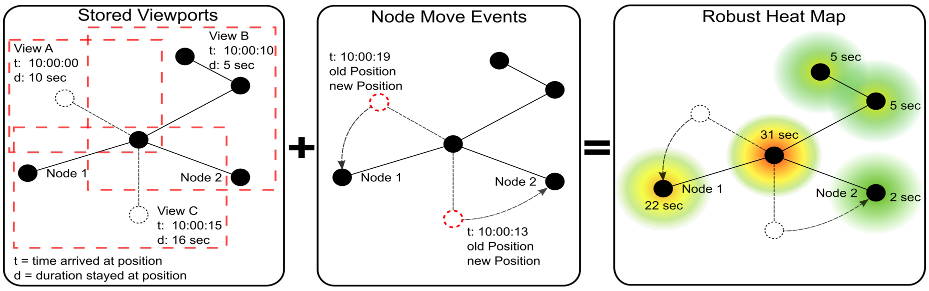


Fig. 3. Illustration for the correlation of all stored viewports with all node move actions to create a heat map that is robust against layout changes of the graph.

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. To solve this issue, users can adjust the settings to filter out these "big views" and only use zoomed-in views to calculate the heat map. Views are also only tracked if the user is actively working on the graph: if a user switches to another window or tab, then the tracking is stopped. It is also stopped if the mouse is not moved for a while (currently 20 seconds) to avoid tracking views of inactive users. This approach does still include nodes in the views that might not have had attention by an active user, but it gives a better 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.

Displaying Graph Changes In the second case, OnGraX calculates values based on changes that have been performed on nodes. Seven actions (name changed, shape changed, node moved, node added, node selected, edge added, edge removed) are tracked and can be used to calculate the heat map values in this case. A multiplier is specified in a configuration dialog for each individual action type to give it more or less weight during the calculation. This enables analysts to highlight only nodes that were moved and had their names changed, for instance. The visualization can be configured to only show a specific user or to show the data for all users together (the selection of user groups would also be possible and could easily be added to the system). Furthermore, it is possible to select a time frame, for instance, the last five minutes 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 specific time frame or to check the work of a single user.

4.3 Tracking and Replaying User Actions

Actions performed by other users during a *synchronous* session are shown at the right corner of the screen (cf. Figure 1, (c)) together with the name of the user who initiated the action. A right-click is used to dismiss a recent action and a left-click moves the camera to the location of the action in the graph. Another left-click on the same action moves the camera back to its original position. Thus, users can quickly check what their collaborators are doing and then return to their own work, without having to navigate to every performed action manually. To provide our users with the possibility to keep track of *all* actions that occurred in a session, we use a scrollable timeline at the bottom border of the screen that shows the complete action history of the graph session (cf. Figure 1, (d)). The mouse tooltip for the symbols in the timeline shows the action time and the name of the user who performed the action. The timeline can also be used to revisit old graph states and replay previous actions. If a user clicks on a symbol, all actions performed since this specific action are replayed in reverse order. The visualization will show the graph in a state before the action was performed. Shortly after the graph has been transformed to its old state, the clicked action is reapplied, animating the graph to the requested point in time. This feature gives users a tool to revisit old graph states and replay old actions allowing them to assess what work has been done by other collaborators. Clicking on the rightmost symbol reverts the graph back to its present state. While viewing an old graph state, it is *not* possible to apply any changes to the graph. We decided against this feature as it would open the possibility to create numerous new branches of different graph states. This is an interesting aspect and actively researched [14], but currently not the focus of our work.

5 Heat Map Evaluation

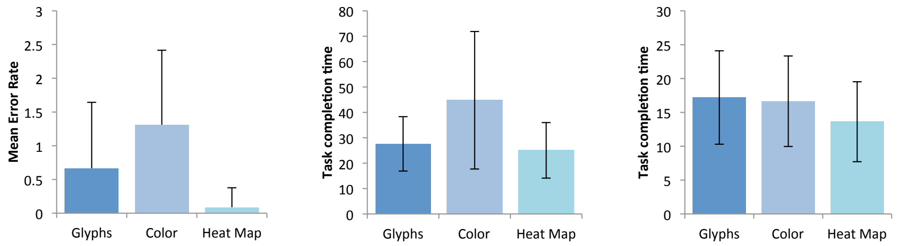
We performed a user experiment to evaluate the usefulness and acceptance of our heat map approach to visualize user behavior data in comparison to glyphs and node coloring. We recruited 15 participants (7 undergraduate students, 7 graduate students, and 1 post-graduate; average age = 28; 5 female, 10 male). Seven participants had a background in computer science and eight a background in media technology. Eight participants never worked with node-link diagrams before, but everyone was familiar with them.

All 15 sessions were recorded on video and the participants were instructed to employ a think-aloud protocol. Before starting the actual tasks, the tool and the three visualization approaches for user behavior data (glyphs, node color, heat map) and their meaning were introduced by the experimenter and each participant could explore a sample graph to get accustomed to the tool. Each session took about 25-30 minutes, and we asked the participants to solve each task as quickly as possible, but the time for the tasks was not limited by us. All participants had to solve two tasks for nine different graphs with the help of the three visualization approaches. Both tasks were described as follows:

Task 1 – explore graph changes: Find and count all nodes that were moved by a specific user (9-14 single marked nodes per graph).

Task 2 – explore viewports: Find all regions that a specific user was most interested in (1-3 marked regions per graph).

The experiment was conducted as a within-participants experiment, and users were divided into three different groups. Every group explored all graphs in the same order but with a different sequence of visualization approaches. Six graphs were generated randomly: the first three graphs consisted of 1,000 nodes/edges and the following three of 2,000 nodes/edges. For the last three graphs, we used existing metabolic networks with 1,300 to 1,800 nodes/edges.



(a) Task 1, mean error rate (b) Task 1, mean completion time in seconds (c) Task 2, mean completion time in seconds

Fig. 4. Analysis of the two tasks for the different visualization approaches.

Quantitative Results We started measuring the task time in seconds for each task as soon as the visualization of the user behavioral data was enabled by the participants and stopped the time as soon as they reported a number. For Task 1, we show the number of nodes that were not found by the users (mean error rate). In Task 2, all participants found all marked regions, regardless of the visualization approach. Therefore, we only report the error rate for Task 1. Figure 4 shows the summarized results for all graphs. Initial Friedman tests showed that both tasks had statistically significant differences in task completion time. Task 1: $\chi^2 = 34.881$, $p < 0.001$. Task 2: $\chi^2 = 16.812$, $p < 0.001$. We conducted a post hoc analysis with Wilcoxon signed-rank tests for our not normally distributed data. For Task 1, the median interquartile range (IQR) task completion times were 26 (Glyphs), 39 (Node Colors), and 20 (Heat Map). Both, glyphs vs. heat maps ($Z = -3.678, p < 0.001$) and node colors vs. heat maps ($Z = -5.334, p < 0.001$) had a significant reduction in task completion time. For Task 2, the median (IQR) task completion times were 19 (Glyphs), 15 (Node Colors), and 13 (Heat Map). Here, the heat map approach also performed significantly better in comparison with glyphs ($Z = -3.678, p < 0.001$) and node colors ($Z = -2.406, p = 0.016$).

Qualitative Results We asked all participants which visualization approach they preferred. Everyone favored the heat map visualization. For them, the heat map was the easiest to perceive, and it also provided the most convenient way to find single nodes with high values, even at lower zoom levels. While performing the second task, four participants mentioned that the glyph approach introduced too much clutter in the view, especially for the metabolic networks. They said that glyphs were hard to distinguish from the actual nodes, because both the nodes and glyphs sometimes had a similar shape.

6 Conclusions

In this paper, we presented a web-based collaborative system for visualizing graphs with several thousands of nodes and edges. Our tool OnGraX provides visualization and interaction techniques for analyzing data sets synchronously *and* asynchronously in a distributed environment. Additionally, 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. We propose using heat maps to efficiently show additional data without affecting the original graph visualization. Based on a user experiment, we show that the heat map-based approach compares better against glyphs or changing the background color of nodes. As future work, we plan to evaluate the other aspects of OnGraX—such as those described in Sect. 4.1 and 4.3—and to use the tool in other contexts. For instance, our collaborators want to use OnGraX for the education of their biology students. The idea is to give students existing metabolic pathways and ask to revise and edit those graphs. Afterwards, the docents could join the online session and discuss those changes with the students. We will use this opportunity to test our tool in another authentic environment and perform a detailed user study during collaborative work in an educational setting. In our specific use case, graph changes are usually limited to a couple of nodes, thus the tracking of *all* actions and visualizing this data is not an issue here. But, it could become problematic if a graph or a subgraph is changed drastically. In this case, additional options to set the granularity for tracked events and alternative visualization techniques would be required incl. a newly designed evaluation.

References

1. KEGG: Kyoto Encyclopedia of Genes and Genomes, `http://www.genome.jp/kegg/`, accessed 12th August 2015
2. Alper, B.E., Henry Riche, N., Hollerer, T.: Structuring the space: A study on enriching node-link diagrams with visual references. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems. pp. 1825–1834. CHI '14, ACM, New York, NY, USA (2014)
3. Chuah, M., Roth, S.: Visualizing Common Ground. In: Proceedings of the International Conference on Information Visualization (IV '03). pp. 365–372. IEEE (2003)
4. Cockburn, A., Karlson, A., Bederson, B.B.: A review of overview+detail, zooming, and focus+context interfaces. ACM Comput. Surv. 41(1), 2:1–2:31 (Jan 2009)
5. Collins, C., Penn, G., Carpendale, S.: Bubble sets: Revealing set relations with isocontours over existing visualizations. IEEE Transactions on Visualization and Computer Graphics 15(6), 1009–1016 (Nov 2009)
6. Dieberger, A., Dourish, P., Höök, K.: Social Navigation: Techniques for Building more Usable Systems. Interactions 7(6) (Nov 2000)
7. Friendly, M.: The history of the cluster heat map. The American Statistician (2009)
8. Gutwin, C., Greenberg, S.: Design for individuals, design for groups: Tradeoffs between power and workspace awareness. In: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work. pp. 207–216. CSCW '98, ACM, New York, NY, USA (1998)
9. Heer, J., Agrawala, M.: Design considerations for collaborative visual analytics. Information Visualization 7(1), 49–62 (Feb 2008)
10. Holten, D., van Wijk, J.J.: A User Study on Visualizing Directed Edges in Graphs. Proceedings of the 27th international conference on Human factors in computing systems - CHI 09 p. 2299 (2009)
11. Isenberg, P., Elmqvist, N., Cernea, D., Scholtz, J., Ma, K.L., Hagen, H.: Collaborative Visualization: Definition, Challenges, and Research Agenda. Information Visualization 10(4), 310–326 (2011)
12. Isenberg, P., Fisher, D.: Collaborative Brushing and Linking for Co-Located Visual Analytics of Document Collections. Computer Graphics Forum 28(3), 1031–1038 (Jun 2009)
13. Isenberg, P., Tang, A., Carpendale, S.: An exploratory study of visual information analysis. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1217–1226. CHI '08, ACM, New York, NY, USA (2008)
14. Landesberger, T.V., Fiebig, S.: Interaction Taxonomy for Tracking of User Actions in Visual Analytics Applications. Handbook of Human Centric Visualization pp. 149–165 (2014)
15. Mark, G., Kobsa, A.: The effects of collaboration and system transparency on cive usage: An empirical study and model. Presence: Teleoper. Virtual Environ. 14(1), 60–80 (Feb 2005)
16. Matejka, J., Grossman, T., Fitzmaurice, G.: Patina: Dynamic heatmaps for visualizing application usage. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 3227–3236. CHI '13, ACM, New York, NY, USA (2013)
17. McKeon, M.: Harnessing the Information Ecosystem with Wiki-based Visualization Dashboards. IEEE Transactions on Visualization and Computer Graphics 15(6), 1081–1088 (Nov 2009)

18. Pohl, M., Schmitt, M., Diehl, S.: Comparing the readability of graph layouts using eyetracking and task-oriented analysis. In: Computational Aesthetics. pp. 49–56 (2009)
19. Slocum, T.A., Mcmaster, R.B., Kessler, F.C., Howard, H.H.: Thematic Cartography and Geovisualization (3rd Edition) (Prentice Hall Series in Geographic Information Science). Prentice Hall, 3 edn. (Apr 2008)
20. Viégas, A.B., Wattenberg, M., Ham, F.V., Kriss, J., Mckeon, M.: Many Eyes: A Site for Visualization at Internet Scale. IEEE Transactions on Visualization and Computer Graphics 13(6), 1121–1128 (2007)
21. Špakov, O., Miniotas, D.: Visualization of eye gaze data using heat maps. Electronics and electrical engineering 2(2) (2007)
22. Wallner, G., Kriglstein, S.: Plato: A visual analytics system for gameplay data. Computers & Graphics 38, 341–356 (2013)
23. Zimmer, B., Kerren, A.: Applying Heat Maps in a Web-Based Collaborative Graph Visualization. Poster Abstract, IEEE Information Visualization (InfoVis '14) (2014)
24. Zimmer, B., Kerren, A.: Sensemaking and Provenance in Distributed Collaborative Node-Link Visualizations. Abstract Paper, IEEE VIS 2014 Workshop: Provenance for Sensemaking (2014)
25. Zimmer, B., Kerren, A.: Harnessing WebGL and WebSockets for a Web-Based Collaborative Graph Exploration Tool. In: Cimiano, P., Frasincar, F., Houben, G.J., Schwabe, D. (eds.) Proceedings of the 15th International Conference on Web Engineering (ICWE '15). Lecture Notes in Computer Science, vol. 9114, pp. 583–598. Springer (2015)