

# Interactive Poster: Table-based Visualization and Interaction for Bipartite Graphs

Hans-Jörg Schulz\*

Mathias John†

Andrea Unger‡

Heidrun Schumann§

University of Rostock, Germany

## ABSTRACT

The visualization of bipartite graphs has only recently gained some interest, as more and more application fields begin to use this special graph class as a natural way to encode their data. We present a table-based approach for the visualization of real world bipartite graphs, which may be large in size and contain a multitude of attributes. To perform complex selections within such graph data, we introduce a new script-based brushing mechanism which integrates naturally with the tabular graph representation.

**Index Terms:** I.3.8 [Computing Methodologies]: Computer Graphics—Applications; H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces

## 1 INTRODUCTION

In graph theory, a graph is called *bipartite* if its set of vertices can be partitioned into two disjoint sets, so that within each of the two sets, there are no adjacent nodes. While bipartite graphs occur frequently in a variety of application fields [1] (as movie-actor-networks, as ontologies, as petri nets, etc.), only recently, their importance has been noticed and first visualization techniques appeared: a well adapted node-link-visualization called *Anchored Maps* [6], specific table-based approaches for visual analytics [8] and transition graphs [?], as well as some solutions that have been specifically designed for movie-actor-networks as in the InfoVis 2007 contest [5]. We adapt the idea of a tabular display and extend it with additional features that allow the technique to scale up to large networks with a multitude of node attributes. Additionally, we propose a new script-based method to interactively query and filter large bipartite graph structures for subgraphs of interest.

## 2 THE VISUALIZATION TECHNIQUE

Using tables to display the node sets comes as a natural choice, as every user being familiar with spreadsheet applications is able to interact with a table instinctively. The two tables representing the two independent node sets form the basis of our visualization. So, each row of these tables represents one node and displays the node attributes. Between these two tables, the connecting edges are shown as lines, which run from a node's row within one table to a row of the other table. Edge weights are mapped onto the width of the individual edges. Additionally, the two 1-mode projections, which basically represent paths of length 2, are depicted as arcs at the side of the two tables. It is apparent that this way of presenting the graph structure is specifically fitted for bipartite graphs, whose edges always connect two nodes from the two different node sets, but never from the same. The overall layout of the described tables, edges, and arcs is shown in Figure 1.

This basic setup of our table-based visualization technique is able to represent all features of a bipartite graph, including derived ones like the projections, in a very compact way. Yet, especially

large and dense bipartite networks demand some additional features to minimize visual cluttering and to enhance the accessibility of our representation. There are basically three problems to target here: large node sets resulting in long tables, a large edge set causing a cluttered display between the tables, and large attribute sets leading to very wide tables. Therefore, the following extensions to the basic layout address these problems caused by large data sets. They are marked by their respective letter with which they are shown in Figure 1.

(A) Using a table lens [7] to reduce the height of the table by minimizing all rows that are not part of the current region of interest. In our case, the region of interest within a table is defined by the position of the mouse cursor.

(B) Adding selection markers to the scrollbars to easily find regions with selected rows in large tables and augmenting the scrollbars with a fisheye distortion to make clicking an individual marker and thus jumping to the respective row easier.

(C) Hiding unselected rows to reduce the current view of the tables to show only selected rows. In this condensed view, unselected rows will be substituted blockwise by a single row that gives information about how many rows have been hidden at that point.

(D) Minimize edge crossings and by this reducing edge clutter. This is done by reordering tables using a barycentric crossing minimization heuristic [4] as the edges running in between the tables just follow the ordering of the rows.

(E) Making edges clickable so that a simple mouse click carries the user right away to an off-screen node without the need to scroll and visually follow the edges which is possibly obscured by other edges.

(F) Using URL-references to out-source secondary node attributes to HTML pages, from where they can be retrieved on demand. A mouse click on a hyperlinked name or label of a node opens up an internet browser window that displays the node's respective HTML page.

However, for further analysis or in-depth visualization of large bipartite networks, the ability to derive a subset of the entire data set is also important. This issue is addressed by our structure-based selection mechanism.

## 3 THE STRUCTURE-BASED SELECTION MECHANISM

While gradual selection concepts for data in general [3] and hierarchies in particular [2] have already been developed, up to this point no selection method specifically tailored to networks is known. Our structural selection method fills this gap by adapting the view of spatial proximity in the graph structure. This means that we are measuring the distance of a node to a focus node within the graph structure itself and not within their screen representation. Thus, two nodes might have been laid out far apart on the screen, but if they are connected by an edge, their structural proximity is still considered high. As we want to make the full power of such a structural selection available to the user, we also strived for increasing its ease of use. As a result, we decided to split the selection process into an interactive and an automatic part. For this, the user selects a set of *focus nodes* in an interactive first step. This triggers the automated part, in a second step, which gathers successively all nodes that can

\*e-mail: hjschulz@informatik.uni-rostock.de

†e-mail: mjohn@informatik.uni-rostock.de

‡e-mail: aunger@informatik.uni-rostock.de

§e-mail: schumann@informatik.uni-rostock.de

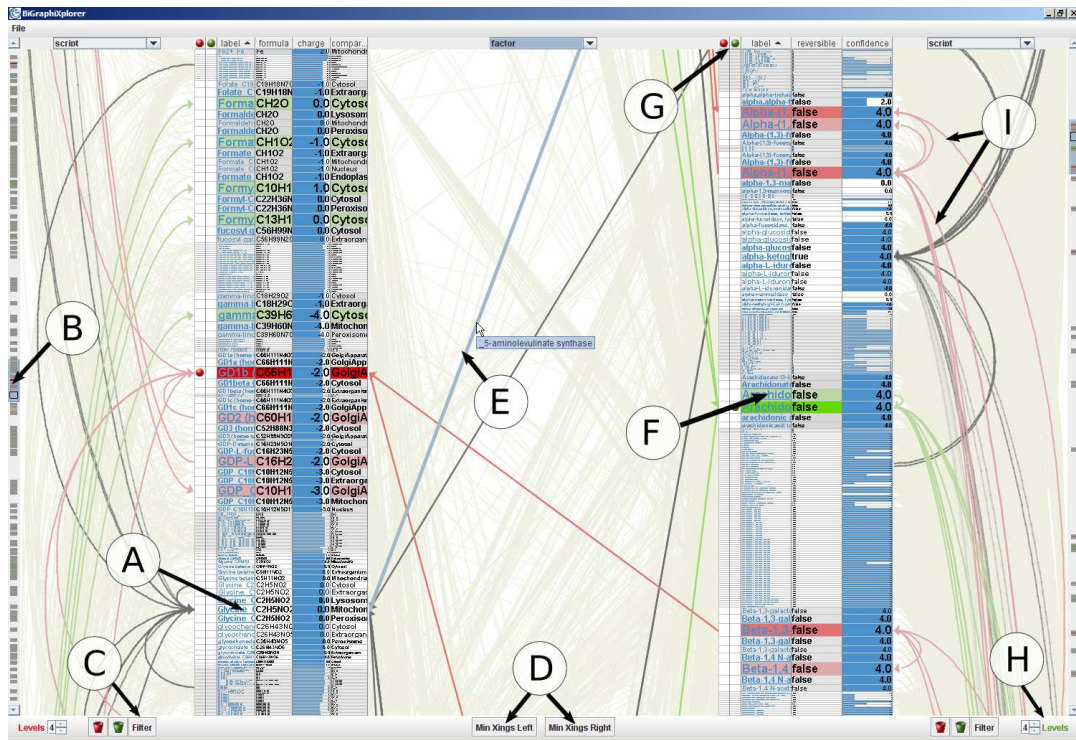


Figure 1: This screenshot shows the two node sets as tables, the connecting edges in between, and both 1-mode projections at the sides. The markers point to the special visualization features we added to the basic concept. (A) - Focus+Context in table, (B) - Fisheye scrollbars with selection markers, (C) - Hide unselected rows, (D) - Minimization of edge crossings, (E) - Clickable edges, (F) - URL-references, (G) - Columns for the two scripts, (H) - Maximum level of script, (I) - Highlighting of traversed edges and 1-mode projections.

be reached by its predefined, set-based selection logic from the set of focus nodes.

**1st Step: The interactive selection.** In our visualization, we chose a toggling mechanism to mark focus nodes, as it can be seen in Figure 1. Additionally, the user can just click'n'drag along the selection columns to select entire regions or intervals of rows. Together with the ability to sort the table with respect to any attribute, this provides a brushing mechanism that allows to quickly select items within certain attribute ranges. To enable the comparison of results from two different selections, two columns are added in both tables (marked by (G) in Figure 1), where focus nodes can be picked independently.

**2nd Step: The automatic selection.** The result of the interactive selection from the 1st step is passed to the loaded selection script. Starting with a set that contains only the focus nodes, the scripted selection logic proceeds to add new node sets script line by script line in a breadth first manner by traversing along edges or projections. The paths along which the script traverses the graph are highlighted in the visual representation of the bipartite graph (marker (I) in Figure 1) The maximum number of selection levels can be set interactively from within the visualization, as shown in Figure 1 at marker (H). Our selection logic allows to define a DoI-value in every script line, which is then assigned to the node set added to the selection at this level. In our table-based visualization, the DoI-values define how the nodes of the set should appear in the tables: it is used to define the color saturation and the height of the respective row.

## 4 CONCLUSION

The presented visualization and selection technique has been successfully tested on several real-world data sets from the life sciences, with some of them containing more than 35.000 nodes and

more than 370.000 edges. So far it appears that, because of the possibility to sort the nodes within the tables by different attributes, the table-based representation outperforms other display techniques for bipartite graphs for localization and comparison tasks. Altogether, the table-based visualization and the script-based selection seem to provide an efficient way of exploring large bipartite networks.

## ACKNOWLEDGEMENTS

This work is supported by the DFG graduate school *DIEM oSiRiS*.

## REFERENCES

- [1] A. S. Asratian, T. M. Denley, and R. Häggkvist. *Bipartite Graphs and Their Applications*. Cambridge University Press, 2004.
- [2] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Navigating hierarchies with structure-based brushes. In *Proc. of IEEE InfoVis*, pages 58–64, 1999.
- [3] G. W. Furnas. Generalized fisheye views. In *Proc. of ACM SIGCHI*, pages 16–23, 1986.
- [4] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, 1(1):1–25, 1997.
- [5] R. Kosara, T. Jankun-Kelly, and E. Chlan. Information visualization contest 2007. In *Proc. of IEEE InfoVis*, 2007.
- [6] K. Misue. Anchored maps: Visualization techniques for drawing bipartite graphs. In *Proc. of HCI International*, volume 2, pages 106–114, 2007.
- [7] R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proc. of ACM SIGCHI*, pages 111–117, 1994.
- [8] J. Stasko, C. Görg, and Z. Liu. Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008.