Visual Analysis of Bipartite Biological Networks

Hans-Jörg Schulz, Mathias John, Andrea Unger, Heidrun Schumann

Eurographics Workshop on Visual Computing for Biomedicine (2008)
Charl Botha and Gordon Kindlmann and Wiro Niessen and Bernhard Preim (Editors)

Visual Analysis of Bipartite Biological Networks

Hans-Jörg Schulz, Mathias John, Andrea Unger and Heidrun Schumann

University of Rostock, Germany

Abstract

In life sciences, the importance of complex network visualization is ever increasing. Yet, existing approaches for the visualization of networks are general purpose techniques that are often not suited to support the specific needs of researchers in the life sciences, or to handle the large network sizes and specific network characteristics that are prevalent in the field. Examples for such networks are biomedical ontologies and biochemical reaction networks, which are bipartite networks – a particular graph class which is rarely addressed in visualization. Our table-based approach allows to visualize large bipartite networks alongside with a multitude of attributes and hyperlinks to biological databases. To explore complex network motifs and perform intricate selections within the visualized network data, we introduce a new script-based brushing mechanism that integrates naturally with the interlinked, tabular representation. A prototype for exploring bipartite graphs, which uses the proposed visualization and interaction techniques, is also presented and used on real data sets from the application domain.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computing Methodologies]: Computer GraphicsApplications; H.5.2 [Information Systems]: Information Interfaces and PresentationUser Interfaces

1. Introduction

In the fields of biology and medicine, large databases have been set up over the last years to allow access to the ever increasing flood of publicly available biological data (http://biodatabase.org). These databases are partially interconnected through cross references, and provide a wealth of information to the research community – as long as they can find it. While many of the databases can easily be queried through special interfaces and some of them even provide static depictions of their data (e.g. drawings of pathways or heatmaps for gene expression data), interactive means of visualizing and querying biomedical data bases are still a current research topic [KS06, SKS07].

In this paper, we present an interactive visualization approach especially designed for bipartite networks [ADH04]. These are graphs that allow their set of vertices to be partitioned into two independent sets, so that within each set, there are no adjacent nodes. Bipartite networks occur frequently in biology and other application domains: e.g. as ontologies with concepts and instances, in modeling and simulation as petri nets, and as biochemical reaction networks.

Only recently, their importance has been noticed and first visualization techniques appeared: a well adapted node-link-visualization called *Anchored Maps* [Mis06] as well as some solutions that have been specifically designed for movie-actor-networks as in the InfoVis 2007 contest [KJKC07]. Yet none of these works are concerned with biomedical data and users from this field.

Our visualization approach has been developed within our research training school *dIEM oSiRiS* (http://diemosiris.de), where biologists, medical researchers, and computer scientists work together to develop modeling and simulation methods as an experimental methodology in biology to achieve new insights into the functioning of biological cell systems. This allows integration of many suggestions and helpful comments from real world users to influence the design process from the very early stages on. During the discussions with the domain experts, two main themes emerged as the most important ones: scalability to large data sets and familiarity to the end users. Because of these two design constraints, we adapted the idea of a tabular display for the node sets. Without cluttering the display, it scales well up to a couple of thousand nodes and even more if focus+context techniques like the Table Lens [RC94] are used. Additionally, every user familiar with spreadsheet applications is able to interact with a table naturally and can focus on learning to handle the additional controls for the explo-

ration of the graph structure. Furthermore, table-based approaches have recently been used in other domains like visual analytics [SGL08] or for the exploration of transition graphs [PvW08]. Yet, these approaches are targeting different application domains which pose different challenges and requirements than the life sciences.

We enhance traditional spreadsheet-based visualizations by showing two tables side-by-side, one for each node set in the bipartite graph, which requires additional screen real estate as well as the handling and intuitive communication of two different focus regions within each table. Also, both tables are connected by the lines of the edge set, which inevitably leads to edge crossings and visual ambiguities. When handling the large node and edge sets of real-world bipartite graphs, our technique must be able to minimize the occurring edge clutter and improve the orientation and navigation of the user within such huge networks. This is achieved by combining numerous interaction techniques (clickable edges, clickable selection markers on the scroll bars, etc.) with a new selection mechanism. This allows the user to trigger and to parameterize automated selection scripts, which are able to encapsulate predefined, complex selection logic.

In Section 2, we present our table-based visualization technique for bipartite graphs, as it forms the basis for the complex structural selection techniques presented in Section 3. To underline the usefulness of our visualization technique in combination with the proposed scriptable selections, Section 4 discusses our software implementation in the context of two large datasets: a reconstruction of the human metabolic network [DBJ*07] from the BiGG Database [SBRG] and a snapshot of the Gene Ontology Database [Gen00] together with the Homo Sapiens annotation from the Gene Ontology Annotation Database [CBDL06]. The last Section 5 concludes our paper and covers future work.

2. The Visualization Technique

A bipartite graph $G(V_1, V_2, E)$ consists of two independent node sets V_1 and V_2 and an edge set E connecting both node sets, containing only edges (x, y) with $X \in V_i$ and $y \in V_{j \neq i}$. At the very basis, our visualization consists of two tables that represent the two independent node sets. Both tables have columns for the attribute set A_i of the respective node set V_i that they are displaying. Each row represents one node and displays the node attributes. The connecting edges between the two node sets are represented 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 P_{V_1} and P_{V_2} are computed by adding all edges (x, z) with $x, z \in V_i$, where $\exists y \in V_{j \neq i}$ so that $\{(x, y), (y, z)\} \subseteq E$. The number of different intermediate nodes y that connect x and z is used as a weight on the resulting projected edge (x, z). These projections are basically additional edge sets that connect only nodes of either V_1 or V_2 and can be seen as shortcuts that directly show dependencies between nodes of the same set. They are depicted as arcs at the side of the two tables. Apparently, 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.

The 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 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 result in very long tables, which are tedious to browse and navigate. This is especially true for our two tables which are linked by edges, as the user who is investigating the graph's structure is not browsing the tables sequentially, but is rather following edges from one side to another. This results in a lot of back and forth browsing within both tables, which is more time consuming as tables get larger.
- Large edge sets result in a lot of lines running in between the two tables, which in turn produces a lot of edge crossings. This makes it harder to follow the course of an individual edge. For this reason, edge crossings are usually to be avoided by any standards of graph drawing aesthetics.
- Large attribute sets with dozens of attributes per node, as they occur in real world data, would result in very wide tables with equally dozens of columns. Since we do not want to introduce another scrolling axis by allowing horizontal scrolling, the visualization is limited in its width by the current screen width, whereas it can be arbitrarily extended downwards. The situation gets even worse if node attributes consist of textual descriptions or even images and figures, as it is not uncommon for real world data sets.

Therefore, the following three sections give an overview of additional features which have been included to reduce visual clutter and to maintain the orientation of the user.

2.1. Additional Features for Large Node Sets

Focus+Context (A): Using a focus+context technique like the table lens [RC94] reduces 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. The row under the cursor, as well as its neighboring rows, will be zoomed and can then be read and investigated, as demonstrated in the accompanying video. The reduction of the row height in the context area now allows for faster scrolling even in large tables.

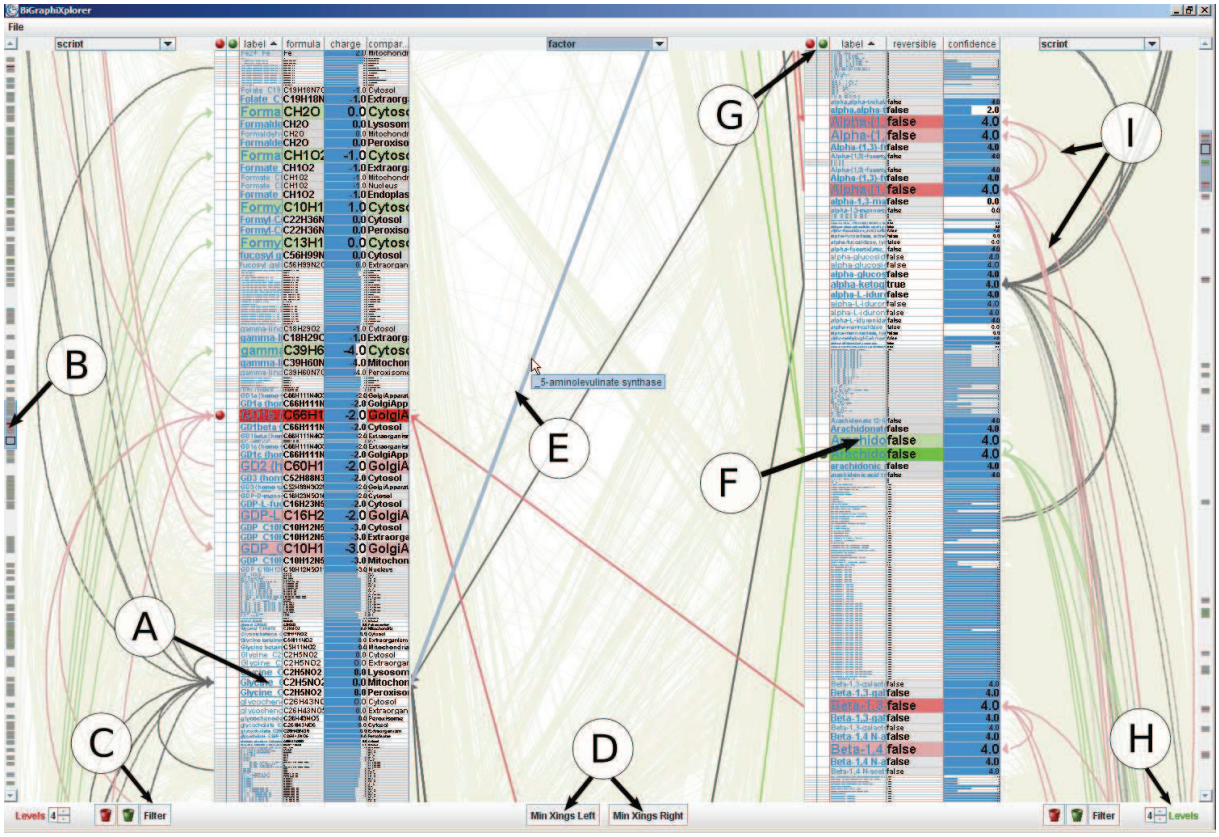

Figure 1: *This screenshot shows the two node sets as tables, the connecting edges in between, and both 1-mode projections at the sides. The markers point to the special visualization features we added to the basic concept. (A) - Focus+Context in table, (B) - Fisheye scrollbars with selection markers,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

Fisheye scrollbar with selection markers (B): As rows of interest can also be selected (see Section 3), such selections can span over both of the tables and be scattered all over them. To easily find regions with selected rows in large tables, we have integrated additional selection markers into the scrollbars at the sides. They indicate where selected rows are located in a table. The user can either use the scrollbar to scroll up/down to a selection or directly click on a selection marker to jump instantly to the respective row. Because the selection markers can be placed quite densely and are hard to pinpoint for clicking, we have also added a fisheye lens to the scrollbar. This lens follows the mouse cursor and spreads out the focus area so that, even in crowded regions, individual selection markers can be clicked. A tooltip displays information about the row to which a selection marker belongs. This feature is also shown in detail in Figure 2.

Hide unselected rows (C): In very large data sets, even with the help of the selection markers, the exploration of a scattered selection can be tiresome. Therefore, we allow the user to reduce the 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. An example is given in Figure 2, where only the selected rows of a table are shown.

2.2. Additional Features for Large Edge Sets

Minimization of edge crossings (D): As the node sets in the tables can be freely ordered, the edges running in between the tables just follow the ordering of the rows. Thus, for a minimization of edge crossings, at least one of the tables needs to be reordered. A barycentric crossing minimization heuristic [JM97] can be called with just one mouse click to rearrange one of the tables. This immensely reduces the visual clutter in most real-world cases.

Clickable edges (E): The visual tracking of edges is hampered by their crossings. This makes it hard to discern the

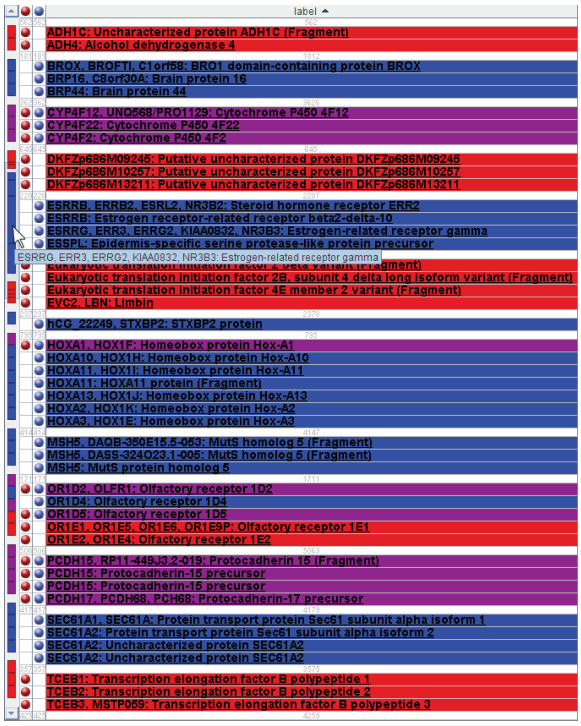

Figure 2: *A table in which unselected rows are collapsed into placeholder rows. The example also illustrates the possibility to adapt the color scheme to the user's personal preferences.*

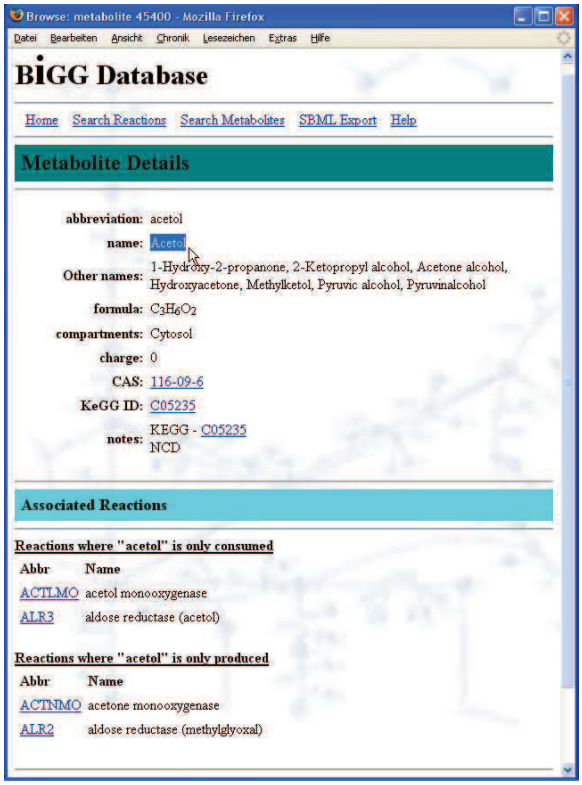

Figure 3: *Linked HTML page - the hyperlinked name of a focussed node was clicked such that a browser starts showing an HTML page with secondary attributes.*

edges and increases the cognitive load for the user. Thus, it is possible to move the mouse over an edge of interest that in turn gets highlighted and becomes clickable if one of its end nodes lies outside of the screen. Additional information on the off-screen endpoint is displayed in a tooltip and a simple mouse click carries the user to the off-screen node.

2.3. Additional Features for Large Attribute Sets

URL-references (F): If the set of node attributes is large, it makes sense to divide it into primary, relevant attributes and secondary, supplementary attributes. Primary attributes are shown in the table where the columns can be resized to provide more space in case of lengthy attribute data and tooltips provide quick access to even more details on demand. Secondary attributes on the other hand are outsourced to HTML pages, from where they can be retrieved on demand. A mouse click on the hyperlinked name or label of a node opens up an internet browser window that displays the HTML page (see Figure 3). The choice of the secondary attributes is constrained by the following two requirements:

- Even with the reduced set of primary attributes, it should be possible for the user to identify individual nodes. So, any name or label that is part of the attribute set should be a part of the primary attribute set.
- The primary attribute set should still allow to perform the exploration in the desired way: e.g., if a comparison of nodes with respect to some attribute is to be performed, this attribute should be part of the attributes shown in the table.

3. Structure-based Selection Mechanisms for Bipartite Networks

Especially in large bipartite networks, the ability to derive a subset of the entire data set is important for further analysis or in-depth visualization. Mechanisms for deriving such a subset can be categorized as:

Addition vs. deletion: These are the two dual modes of constructing a subset. One can either start with an empty subset and add the desired data values, or with the complete data set and consecutively delete undesired data values. For interactive selection, often both modes are combined to allow for the refinement of a previously constructed subset. The Toggle-functionality can be seen as the simplest example for that, but also more complex mechanisms have been shown to

be useful [Wil96]. Our structure-based selection mechanism makes use of selection by addition. This is mostly motivated by the fact that it is usually easier to define what kind of data is of interest, instead of defining its inverse.

Automatic vs. interactive: The subset can either be constructed automatically or be specified interactively by the user. In visualization, selection is generally understood as the interactive counterpart to the more automatic, preprocessing-oriented sampling (constructed by addition) and filtering (constructed by deletion). In this section, we present a 2-step selection mechanism for bipartite graphs. Its two steps divide the selection process into an interactive and an automatic part, which allows to combine the best parts of both worlds: the possibility for the user to influence the selection by interactively triggering and parameterizing a complex automated selection logic.

Binary vs. discrete vs. continuous: In the binary case, a data value either belongs to a subset or not. Whereas in the discrete and continuous cases, each data value is mapped to a Degree of Interest (DoI) between 0 and 1 [DH02], which determines "how much" a data value belongs to the subset. Discrete and continuous selection methods come naturally with focus+context techniques. Here, the focus region belongs to the selection and is assigned a DoI of 1. Whatever lies outside of this focus region is mapped to a DoI between 0 and 1. When defining such a mapping, there is usually some notion of relatedness to the focus region involved. This relatedness can simply be spatial proximity in data or image space [Fur86], or more complex semantic concepts like similarity [NH06] or relevance [SCH*06]. The selection mechanism presented in this section uses a discrete DoI based on the user's interest in specific parts of the subset.

3.1. Combining Interactive and Automatic Selection

Accounting for structural relatedness, selection mechanisms have already been presented for trees and hierarchies [FWR99]. There, they are implemented as a highly interactive brushing mechanism. A number of different parameterizations allows to refine the scope of the selection within the tree. As we now take structural selection further to the more complex case of bipartite graphs, the number of possible configurations in the selection scope becomes even larger. This is because there is more than one way of being related in a bipartite graph: being adjacent through the edge set E, being adjacent through one of the 1-mode-projections P_{V_1} or P_{V_2}, or even not being directly adjacent, but still being connected through a path of two other nodes. Our structural selection method adapts the view of spatial proximity in the graph structure as shown by these examples. 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.

In order to allow the full power of these configurations without reducing the ease of use of the selection, 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 is passed to the automated part, in a second step, which gathers successively all nodes that can be reached by its predefined, set-based selection logic.

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 and 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. A unique color is assigned to each of the two individual selections and everything related to them (icons, menu entries, highlighted rows and edges, etc.). By default, red and green are used, as biologists are familiar with this color scheme. If it happens that a node is affected by both selections, the color yellow is used for its respective row in the table, which is inspired by traffic lights, where yellow lies between red and green. However, the colors can be adjusted according to the user's preferences.

2nd Step: The automatic selection. The result of the interactive selection in the first 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 maximum number of selection levels is determined by the number of lines within the selection script. It can be interactively lowered from within the visualization for both available scripts, 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 by the script line. That way, the user's interest in the result of individual script lines can be specified more precisely compared to an automatically derived DoI-value, for example by proximity to the focus nodes. The values lie between 0 and 100, where 0 means no accentuation or amplification of a row, and 100 means maximum accentuation. The value ranges have been chosen because we found that the "percentage-thinking" is intuitively understood throughout all application domains. 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. An example of such a selection script is given in the context of the examples discussed in Section 4.

To make our concept more flexible, we extended it to allow the filtering of node sets by the nodes' attributes values. These can be attributes from the dataset itself or some

of the commonly derived structural attributes, like indegree, outdegree or the clustering coefficient, which is used in its adapted version for bipartite graphs [RA04]. The filtering of node sets can be used to narrow down the traversed paths to only those paths of interest. 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 edges along the paths are colored with the same saturation as the nodes they emanate from. This is very helpful for understanding the inner workings of selection scripts, as one is not only given the script result (nodes with DoI-values attached), but also some information on how this result came about. It ties in very nicely with the possibility to interactively define the maximum level up to which a script is computed.

4. Discussion

This section will give an overview of the usage of the visualization and selection technique by discussing two real world data sets. The first data set, a biomedical ontology, is foremost used to describe the visualization technique, and the second data set, a biochemical reaction network, is then used to illustrate the selection mechanism.

4.1. A Biomedical Ontology

Ontologies are a collection of interrelated concepts or terms that aim at providing a basis for knowledge management in a certain area. Data sets can use the provided terms, to declare their data instances of these concepts, annotating them and thus providing semantics alongside the pure numbers. These annotations form a bipartite graph, with the terms as one node set and the instances as the other. This bipartite model is well known from research on Semantic Web technologies like folksonomies or lightweight ontologies.

One of the largest biomedical ontologies freely available today is the Gene Ontology Database [Gen00] (OBO snapshot from 12-JUN-2008), which we used together with one of the largest annotated data set of gene products, the Homo Sapiens annotation from the Gene Ontology Annotation Database [CBDL06] (GOA snapshot from 07-JUN-2008). This data set consists of 26389 terms from the ontology and 35043 gene products annotated with these terms. These annotations are encoded in 377132 undirected links in between these two sets. By projecting these annotations onto the set of terms, 122379 additional edges are produced. The projection onto the gene products ran into a combinatorial explosion and produced literally billions of additional edges. Since this number is well beyond the scope of our tool, we filtered this huge set of projected edges down to the 291586 edges, with weight 16 or higher. The weight of 16 is basically an arbitrary choice, as it presents the lowest filtering threshold for this data set that yields an edge set with less than a million edges, which is more or less the limit of our implementation. The data sets also have a variety of attributes, as short natural language definitions for the terms and bibliographical references for the gene products. To make all of this information available, the terms have been hyperlinked to the Gene Ontology Database and the gene products to the five databases they originally stem from: UniProtKB, RefSeq, ENSEMBL, H-invDB, and VEGA. So, this is a good example of how our tool ties together individual biological databases.

When exploring this rather large data set with our tool, it soon becomes apparent that scrolling through the tables is a very time consuming and slow method to access the network. It actually takes more than 700 turns on the mouse wheel to scroll from the top of the terms table to its bottom. So, this is exactly one of the cases, where all the other means of rapid navigation come into play: clickable edges, fisheye scrollbar and folding of unselected nodes to compress the view. As shown in Figure 2, the folded view of the terms table compresses several thousand unselected rows into one-row-placeholders. Navigating in the compressed view and only switching back to the full view if a new selection is to be made, speeds up the exploration process immensely.

The ontology data set exhibits an interesting way of using the projections: as indicators of similarity. That is because the number of shared terms is often taken as a measure of similarity for two instances. Also, the more instances two terms have in common, the more alike they probably are. This notion is captured exactly by the projections and their weights: the higher the weight of a projected edge, the more disjoint paths exist between its two incident nodes. And the more disjoint paths there are, the more joint intermediary nodes must lie within the other node set. Hence, clicking and thus following the projections during the exploration makes it very easy to find similar nodes within this large data set – something that would not have been possible without the projections on the side.

4.2. A Biochemical Reaction Network

Reaction networks are usually described as hyper graphs, in which each directed hyper edge encodes a biochemical reaction connecting its reactants with its products. We use a common transformation of hyper graphs into the so-called König's representation [TZB96] to map the reaction network to a bipartite graph. Hereby, the hyper edges (reactions) are transformed into nodes themselves and new directed edges are introduced from each reactant to its reaction and from each reaction to its products. This transformation yields two node sets: the chemical substances and the reactions.

As one of the largest biochemical reaction networks that has been reconstructed so far, we used the human metabolic network [DBJ*07] from the BiGG Database [SBRG] (SBML snapshot from 20-DEC-2007). It comprises 2764 chemical compounds, 3311 chemical reactions and 17519 directed links with stoichiometric factors encoded as weights between compounds and reactions. Additionally, the projec-

tion on the compounds yields 121118 edges and the projection on the reactions 295922 edges. The data set includes an abundance of attributes, from charges for the compounds to bibliographical references for most of the reactions. As not all of them could be displayed, the names of the compounds and reactions are hyperlinked to the BiGG database with all the additional information, as seen in Figure 3

To illustrate the script-based selection mechanism, we derived a selection logic for a dependency analysis, as this kind of analysis is often performed by biologists on reaction networks. This means, for a focussed chemical compound the selection script should automatically determine on which other compounds it directly depends. As there are no edges between the nodes of one node set, direct dependence means that there exists at least one directed path of length 2 between them. So, such a script must follow down all reactions that produce this very compound and select their reactants. This selection logic needs to be transformed into a set-based notation, which could look like the following, with $V = V_1 \cup V_2$:

$$Level_0(\text{DoI} = 100\%) = \text{ focus nodes}$$
$$Level_1(\text{DoI} = 0\%) = \{u \in V : \exists (u,v) \in E : v \in Level_0\}$$
$$Level_2(\text{DoI} = 75\%) = \{u \in V : \exists (u,v) \in E : v \in Level_1\}$$

Here, with each step, the logic is traversing backward edges, starting from the focussed compound and traversing backwards twice: once to reach all reactions that produce the compound and another time to reach their reactants. Thereby, it is guaranteed to return in $Level_2$ only nodes that are also chemical compounds. The example also shows that the DoI is not necessarily decreasing with each level. Instead, each level can be assigned its individual DoI according to the user's goal of analysis. In our example, we are only interested in the compounds, but not in the intermediate reactions and assign them therefore a DoI of 0. Given the intended logic in a set-based notation, it can now be written as a selection script, which is shown in Figure 4.

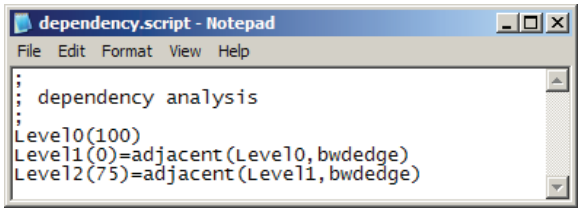

Figure 4: *An example of a selection script.*

Once defined, this script can be loaded and used with a mouse click on whichever compound the biologist likes to run the dependency analysis on. This is the interactive part of the selection, which toggles nodes to build up a set of focus nodes to be passed to the script for dependency analysis.

It would also be possible to shorten the shown selection script by traversing the backward 1-projection on the compounds in just one step, instead of going two steps along the edges themselves as before. This illustrates nicely what we meant when we introduced the 1-mode-projections as shortcuts within the bipartite graph. It is, of course, possible to create much more complex selection scripts, as we have done for the example in Figure 1, which shows a part of the discussed metabolic network.

5. Conclusion and Future Work

We have presented a compact visualization technique for the interactive exploration of large bipartite biomedical networks that adapts the intuitive and familiar tabular display to the characteristics of bipartite graphs. Displaying the two node sets in separate tables offers a natural way to lay out the bipartite graph and to combine structural information (edges and 1-mode projections) and multiple attributes in one display. Furthermore, the tabular display is scalable to very large data sets and provides features to easily rearrange the node sets based on their attribute values. These are major advantages of our concept in comparison to standard node-link graph layouts with their cumbersome node placement and label arrangement. A rich set of interaction techniques allows for a convenient exploration of the bipartite graph, e.g. lens functions to provide focus+context and functions to jump directly to interesting parts of the data outside of the display. Another important aspect in this regard is the combination of interactive and automated selection of nodes: an interactively defined selection is extended by automatic, script-based processing. While the technique in its whole is tailored to the concrete application domain, the underlying concepts, e.g. the multi-tabular display, the edge-based navigation, or the selection scripts can be applied to the general area of visualizing multipartite graphs.

The applicability of our technique to real world data was ensured by including the feedback of domain experts from biology and medical sciences into the design process. Their overall impression of the resulting visualization and interaction techniques, as we have presented them in this paper, was mostly positive. They considered the tabular display an intuitive way to analyze the data and found the 2-step structural selection useful to find features of interest after some time of getting used to its concept.

In addition to the features already present, they would have wished for an even tighter integration with the databases in the sense that navigation through the database interfaces links back and adapts the visualization likewise. This would give them all the power of the visualization while still being able to use the textual query interfaces of the databases as they are used to do. Such a "hybrid" approach will probably also further the acceptance of our tool, as it would result in a more moderate learning curve.

As for enhancements to the visualization itself, we plan to improve the navigation along edges as this has turned out to be the preferred way of the biologists to traverse the network. Here, the problem is, that edges are often collinear and

cluttered, which makes it hard to pick and click a specific edge for traversal. This effect can be reduced by adding edge lenses to locally declutter the view [WCG03, TAvHS06] and allow a precise selection of individual edges.

Another hurdle to overcome is the initial skepticism towards the selection scripts by new users, as their concept seems very abstract at first. Yet, we experienced that after a few minutes of training, this attitude changes. Therefore, we imagine to construct a whole library of different selection scripts and to investigate mechanisms that provide a suitable subset of this library for different exploration goals and data domains. These scripts would provide a good starting point for new users, who could readily use them without having to worry about their inner workings. As scripts can be shared, it is also possible for more advanced users to encapsulate their selection logic and pass their self-made scripts on to others.

Acknowledgements

The authors wish to thank Steffen Hadlak for his work on the implementation and the video presentation. This work is supported by the DFG graduate school *dIEM oSiRiS*.

References

[ADH04] ASRATIAN A. S., DENLEY T. M., HÄGGKVIST R.: *Bipartite Graphs and Their Applications*. Cambridge University Press, 2004.

[CBDL06] CAMON E., BARRELL D., DIMMER E., LEE V.: The gene ontology annotation (GOA) database. In *In Silico Genomics and Proteomics: Functional Annotation of Genomes and Proteins*, Mulder N., (Ed.). Nova Science Publishers, 2006, pp. 37–54.

[DBJ*07] DUARTE N. C., BECKER S. A., JAMSHIDI N., THIELE I., MO M. L., VO T. D., SRIVAS R., PALSSON B. Ø.: Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proc. of the National Academy of Sciences of the USA 104*, 6 (2007), 1777–1782.

[DH02] DOLEISCH H., HAUSER H.: Smooth brushing for focus+context visualization of simulation data in 3d. In *Proc. of WSCG* (2002), pp. 147–154.

[Fur86] FURNAS G. W.: Generalized fisheye views. In *Proc. of ACM SIGCHI* (1986), pp. 16–23.

[FWR99] FUA Y.-H., WARD M. O., RUNDENSTEINER E. A.: Navigating hierarchies with structure-based brushes. In *Proc. of IEEE InfoVis* (1999), pp. 58–64.

[Gen00] GENE ONTOLOGY CONSORTIUM: Gene ontology: tool for the unification of biology. *Nature Genetics 25*, 1 (2000), 25–29.

[JM97] JÜNGER M., MUTZEL P.: 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications 1*, 1 (1997), 1–25.

[KJKC07] KOSARA R., JANKUN-KELLY T., CHLAN E.: Information visualization contest 2007. In *Proc. of IEEE InfoVis* (2007).

[KS06] KLUKAS C., SCHREIBER F.: Dynamic exploration and editing of KEGG pathway diagrams. *Bioinformatics 23*, 3 (2006), 344–350.

[Mis06] MISUE K.: Drawing bipartite graphs as anchored maps. In *Proc. of APVIS* (2006), pp. 169–177.

[NH06] NOVOTNÝ M., HAUSER H.: Similarity brushing for exploring multidimensional relations. In *Proc. of WSCG* (2006), pp. 105–112.

[PvW08] PRETORIUS A. J., VAN WIJK J. J.: Visual inspection of multivariate graphs. *Computer Graphics Forum* (2008), 967–974.

[RA04] ROBINS G., ALEXANDER M.: Small worlds among interlocking directors: Network structure and distance in bipartite graphs. *Computational and Mathematical Organization Theory 10*, 1 (2004), 69–94.

[RC94] RAO R., CARD S. K.: The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proc. of ACM SIGCHI* (1994), pp. 111–117.

[SBRG] SYSTEMS BIOLOGY RESEARCH GROUP UNIVERSITY OF CALIFORNIA S. D.: BiGG: Database of biochemically, genetically and genomically structured genome-scale metabolic network reconstructions. http://bigg.ucsd.edu.

[SCH*06] SUN X., CHIU P., HUANG J., BACK M., POLAK W.: Implicit brushing and target snapping: Data exploration and sense-making on large displays. In *Proc. of AVI* (2006), pp. 258–261.

[SGL08] STASKO J., GÖRG C., LIU Z.: Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization 7*, 2 (2008), 118–132.

[SKS07] STREIT M., KALKUSCH M., SCHMALSTIEG D.: Interactive visualization of metabolic pathways. In *Poster Compendium, IEEE InfoVis* (2007).

[TAvHS06] TOMINSKI C., ABELLO J., VAN HAM F., SCHUMANN H.: Fisheye treeviews and lenses for graph visualization. In *Proc. of IEEE IV* (2006), pp. 17–24.

[TZB96] TEMKIN O. N., ZEIGARNIK A. V., BONCHEV D.: *Chemical Reaction Networks: A Graph-Theoretical Approach*. CRC Press, 1996.

[WCG03] WONG N., CARPENDALE S., GREENBERG S.: Edgelens: An interactive method for managing edge congestion in graphs. In *Proc. of IEEE InfoVis* (2003), pp. 51–58.

[Wil96] WILLS G. J.: Selection: 524,288 ways to say "this is interesting". In *Proc. of IEEE InfoVis* (1996), pp. 54–60.