A Novel Grid-based Visualization Approach for Metabolic Networks with Advanced Focus&Context View

Markus Rohrschneider[1], Christian Heine[1], André Reichenbach[1], Andreas Kerren[2], and Gerik Scheuermann[1]

[1] University of Leipzig, Department of Computer Science, Germany
[2] Växjö University, School of Mathematics and Systems Engineering, Sweden

Abstract. The universe of biochemical reactions in metabolic pathways can be modeled as a complex network structure augmented with domain specific annotations. Based on the functional properties of the involved reactions, metabolic networks are often clustered into so-called pathways inferred from expert knowledge. To support the domain expert in the exploration and analysis process, we follow the well-known Table Lens metaphor with the possibility to select multiple foci.
In this paper, we introduce a novel approach to generate an interactive layout of such a metabolic network taking its hierarchical structure into account and present methods for navigation and exploration that preserve the mental map. The layout places the network nodes on a fixed rectilinear grid and routes the edges orthogonally between the node positions. Our approach supports bundled edge routes heuristically minimizing a given cost function based on the number of bends, the number of edge crossings and the density of edges within a bundle.

1 Introduction

To fully comprehend and appreciate the existing knowledge on chemical processes in living organisms it is essential to develop suitable tools to explore and navigate through vast amounts of information stored in biological databases. In biochemistry, complex networks defined by interactions and relations between different chemical compounds are considered as pathways, such as regulatory pathways controlling gene activity or metabolic pathways comprising chemical reactions for synthesis, transformation and degradation of organic substances in biological systems.

In this work, we combine and apply information visualization techniques to present the complete set of biochemical reactions of metabolic pathways in a eucaryotic cell supplying means of exploration and navigation. Although the emphasis of this paper is placed on biochemical network data, the presented application is not limited to this area. Instead, it can handle any large graph carrying arbitrary annotational information by mapping given data properties to attributes being visualized by the software. To capture the complex chemical interactions of such a reaction network, metabolic pathways may be modeled as

In Proceedings of the 17th International Symposium on Graph Drawing (GD '09), pages 268-279, volume 5849 of LNCS, Chicago, USA, 2009.

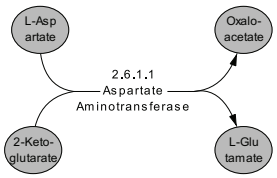

Fig. 1. A hyperedge depicting a transaminase reaction, which converts amino acids into corresponding alpha-keto acids and vice versa. In this example, the enzyme Aspartate Aminotransferase converts the substrates L-Aspartate (amino acid) and 2-Ketoglutarate (alpha-keto acid) [input nodes] into the products Oxaloacetate (alpha-keto acid) and L-Glutamate (amino acid) [output nodes]. Many reactions are reversible, so the direction of the hyperedge simply gives a hint on the reaction's chemical equilibrium.

hypergraphs, where unlike a regular graph, each edge can connect an arbitrary number of nodes. In this hypergraph model, each substance is represented by a node of the graph, and each reaction by a (directed) hyperedge connecting the input node set—substrates—with the output node set—products— of the chemical reaction (see Fig. 1). To obtain a hierarchical graph, each metabolic pathway is represented by a node at the top level, where the pathway's reaction network constitutes the nested graph at the bottom level. The division into separate pathways, although based on expert knowledge, is somewhat arbitrary and may not be a strict partition of the graph. Nevertheless, we consider the clustering of the node and edge set as a partition to obtain a strictly confined hierarchy on the graph. Compound nodes and reactions belonging to more than one pathway are simply duplicated for the sake of simplicity of the resulting graph. This step has two benefits: layouting the graph will be a much simpler task, and we can use the hierarchy to explore the network in a top-down manner by examining the top-level graph at first and adding additional information on pathways of interest by expanding nodes.

2 Related Work

The visualization of large and complex biological networks is one of the key analysis techniques to cope with this enormous amount of data. Here, the layout of networks should be in agreement with biological drawing conventions and draw attention to relevant system properties that might remain hidden otherwise [14, 13]. Further important issues are the preservation of the so-called mental map [1] when applying small changes to the graph and the possibility of clustering nodes. Depending on the concrete network drawing, there are further important visual representation and interaction techniques that play important roles, e.g., navigation in the complete network, focusing on parts of the network, or gradual differentiability of nodes with less importance (side metabolites) [12]. However, only little research has been done in the past to solve the special layout and visualization problems arising in this area. A lot of the most used software systems for the visual analysis of generic biological networks, i.e., different kinds of

networks like regulatory networks or protein-protein interactions, only provide implementations of standard graph drawing algorithms, such as force-directed or hierarchical approaches [8].

Cytoscape [5] is one of the most popular tools for generic biochemical network visualization and supports a number of standard graph layout algorithms. Filtering functions are provided to reduce network complexity. For instance, the user can select nodes and edges according to their name and other attributes. This system allows a simple mapping of data attributes to visual elements of nodes and edges. VisANT [4] is another system designed to visualize generic biochemical networks. In addition to the features of Cytoscape, it provides statistical analysis tools, e.g., based on node degrees or the distribution of clustering coefficients. Their results are displayed in separate views, such as scatter plots.

Especially for metabolic networks, large and hand-drawn posters were produced in the past, for example, Nicholson's pathway map [9] or the widely-used metabolic pathway poster published by Roche Applied Science [18]. Other projects have created graphical representations of metabolic networks and offer them via web pages (e.g., the BioCyc collection [7]). The widespread pathway drawings of the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [16], see also Section 3, were also produced by hand. These drawings are connected via links, but real interaction is not available. Because of their manual generation, they are well readable and can thus serve as an example in terms of quality and user conventions. Moreover, the availability of these representations has established a de facto standard for metabolic network drawings: it features near-orthogonal drawings where, for example, important paths are aligned or relevant subgraphs are placed close to the center of the drawing [14, 13].

Newer approaches are based on a close interdisciplinary work between researchers in visualization and biochemistry. An example is the Caleydo framework [21] that extends the standard pathways of KEGG into 2.5D, similar to the report of Kerren [12] and the work of Brandes et al. [15], combined with brushing, highlighting, focus&context, and detail on demand. In this way, it supports the interactive exploration and navigation between several interconnected (but static!) networks.

Saraiya et al. [19] discussed the requirements of metabolic network visualization collected from interviews with biologists. They observed five requirements that are important for biologists working on pathway analysis, but still not completely realized in existing visualization systems (adapted from [10]):

1. automated construction and updating of pathways by searching literature databases;
2. overlaying information on pathways in a biologically relevant format;
3. linking pathways to multi-dimensional data from high-throughput experiments, such as microarrays;
4. overlooking multiple pathways simultaneously with interconnections between them; and
5. scaling pathways to higher levels of abstraction to analyze effects of complex molecular interactions at higher levels of biological organization.

Currently, our approach addresses several of the aforementioned requirements and improves the most previous work by using of interaction techniques from information visualization. Our new, interactive layouts are based on the KEGG data (Req. 1), and we provided the visualization with an intuitive focus&context view. In this way, we can handle, for example, the *complete* metabolism of a generalized eucaryotic cell (Req. 4) by following Shneiderman's mantra [20]: overview first, zoom and filter, details on demand. If the user explores the pathways interactively, the visualization approach preserves the mental map. To the best of our knowledge, no other system can provide that to this extent. Our system is also able to embed textual information into the drawings and to use glyphs/icons for the representation of lower-level subgraphs if needed, similar to the Pathway tools [3]. The integration of more complicated attributes as well as biological patterns regarding topological substructures are still missing. Here, other tools, such as BioPath [6], still have an advantage to be fully accepted by biologists.

The generation of the actual layout of the hierarchical pathway graph is motivated by the style of the "official" KEGG diagrams to be consistent with the domain experts expectations. The diagrams usually use the orthogonal style for drawing edges. To avoid overlapping labels, we ensure a minimum separation of the diagrams elements by using a regular grid based approach. Algorithms for orthogonal grid drawing have been widely studied; we cannot provide an extensive overview here and refer the reader to [8, 26] for an introduction. These approaches often follow a topology-shape-metrics approach [25]: First, compute a planar embedding of the input graph, possibly planarizing it by augmenting vertices at crossings, second, compute an orthogonal representation of the embedding, and finally generate coordinates by compaction of the orthogonal representation. Usually, edges are not allowed to run simultaneously on the same grid segment, i.e., connection between two neighboring grid positions. The pathways of the KEGG database can be converted into graphs, but a planarization of them requires an enormous amount of augmented vertices. If edges are not allowed to run on the same grid segment, their layout dominates the area of the drawing resulting in poor resolution. Furthermore, as a pathway constitutes a semantic entity, they should be presented as a unit and without diagram elements from foreign pathways interfering. No existing orthogonal drawing algorithm was able to take these constraints into account, therefore we developed our own that does not planarize the graph but keeps track of edge crossings and heuristically minimizes them, allows "edge bundling" although it penalizes it, shows pathways as units and performs dynamic compactions based on the currently focused parts of the pathway hierarchy.

3 Network Data Source

The development of graph interaction techniques especially suited to fit biological problems makes it necessary to experiment with realistic datasets. To generate artificial graph data is of course possible, but it is hard to estimate the required complexity of such datasets to simulate realistic scenarios. The Kyoto

Encyclopedia of Genes and Genomes (KEGG, [16]) System provides annotated pathway data facilitating the construction of metabolic pathway graphs of different sizes. KEGG is one of the major bioinformatics resources publicly accessible. It integrates genomic, chemical, i.e., molecular, and systemic functional information describing cellular processes and organism behavior. It provides a knowledge base for systematic analysis in bioinformatics research and the life sciences. We extracted the hypergraph structure including semantic information as discussed in [17]. The constructed graph covers the complete metabolism of a generalized eucaryotic cell and contains 4980 compound and 154 pathway nodes, 4943 reactions and 1248 inter-pathway edges.

4 Hierarchical Orthogonal Grid Layout

A *hypergraph* $H = (V, E)$ as an extension of the graph concept allows the elements of E called *hyperedges* to connect multiple vertices. Conceptually, a chemical reaction can be described as a hyperedge between compounds that are modeled as vertices. This requires a mark whether a vertex is a substrate or product. We model the data in the KEGG database as a hierarchy of one top-level graph that contains a vertex for each pathway and one hypergraph per pathway. If two pathways exchange compounds according to KEGG, both a regular edge exists between them in the top level graph as well as an edge between the two hypergraphs representing the two pathways.

The layout of the hierarchical KEGG hypergraph is generated by converting the hypergraphs of the hierarchy into their corresponding bipartite graphs and computing a layout of this graph hierarchy. The generated layout is orthogonal to match the style of the official KEGG diagrams. Furthermore, its vertices' positions lie on a grid to ensure both a minimum separation between labels and to make the algorithm both simpler and faster. The layout algorithm allows multi-edges but no loops and proceeds recursively—parents before their children. For each graph of the hierarchy, the layout consists of three phases named: Vertex Position, Edge Routing, and Edge Bundling.

In the Vertex Position phase, we try to find a unique integer position for each vertex that minimizes the stress: the amount of error that takes place by the projection of the "high-dimensional" graph-theoretic distances to the geometric distances between the vertices positions. As vertices and edges are laid out on a regular grid, the Manhattan distance is used as geometric distance. When an edge leads outside the graph, its hierarchy parent has already been laid out. Thus, the direction from which the edge enters the graph is known. For each of the four orientations, we temporarily add a *port* vertex to the graph and connect the edges to foreign graphs to that port. Unlike the other graph's vertices the position of ports is fixed on the boundary of the grid in the upcoming optimization phase.

We implemented a stress minimization algorithm inspired by Kamada and Kawai [22]: starting from an inital random integer positioning of vertices, we select a vertex with high local stress and find a continuous position for that vertex where its local stress becomes minimal using the Newton-Raphson method. We insert

it then at the closest integer position not taken by any vertex. We compared this method with two different approaches. A *brute force* version picks a random vertex and tests all integer positions in a vincinity of its position and insert it at the best position. A *simulated annealing* [23] variant picks a random vertex and new position in the vincinity of the old position, but performs the insertion only on improvement and deterioration with decreasing likelihood. We found that the *brute force* method optimized quality as it is difficult to trap this method in a local minimum. The *simulated annealing* is very fast, but does not provide nearly the same quality. All heuristics terminate after a fixed number of iterations that is proportional to the number of vertices.
The Edge Routing phase computes a combinatorial description of an edge routing along the edges of the regular grid. The vertices' positions are not altered by this phase. The combinatorical description is computed one edge at a time and after all edges have been processed once, an iterative process removes single edges and adds them again optimizing on the global cost of the layout. Given a combinatorial description of the current edge routing, we construct a *route graph* that consists of the original graph's vertices and the grid's edges as vertices and edges for valid transitions between these elements. Given this representation, we are able to compute the optimal routing of an edge by solving a single-pair-shortest-path instance on the *route graph*. The optimality is given by a cost function that takes the number of crossings, the number of bends, the length of an edge, and the "density" of edges on a grid segment into account. Note that the quality of the resulting configuration depends both on the original vertices' positions and the actual order of edge insertions. Good performance was achieved when inserting the edges in the order of increasing distances of their incident vertices. To reduce runtime and memory consumption, we use the A^* search algorithm [24] to solve the SPSP instance using the Manhattan distance as heuristic.

The Edge Bundling phase shifts segments of edges' routes orthogonal to the grid segments they lie on to remove overlaps. It preserves the edges' relative ordering and straightens them in the process. This problem can be solved for each row and each column separately. We generate for each row and each column a directed acyclic graph that contains line segments as vertices and edges between these lines, if they are ordered in the combinatorial edge routing. Any topological numbering of this graph gives a displacement that avoids occlusions between edge routes of the same column/row, and using the topological numbering of minimum weight packs the edge bundles nicely together.

5 Graph Interaction

The graph interaction and exploration methods described in this section have all been implemented in our visualization software. The grid layout algorithm is the central component of the adapted Table Lens method to explore hierarchical graphs. We firstly present this technique with supplementary search and highlighting operations and explain later how the graphical user interface lets the user apply these methods to interact with the metabolic network graph.

5.1 Exploration Techniques

Two fundamental navigation operations on hierarchical graphs are node expansion to reveal the node's nested graph and collapse. For 2D graph representations, it is natural and desirable to present a flat graph at all times regardless the graph's expansion state. This means that the expansion of a node requires it to be hidden and replaced by its nested graph. The inverse operation replaces the nested graph by its parent. The well-known Table Lens metaphor [2] applied to hierarchical graph exploration fulfills this requirement. It is an established focus&context method to give an overview on large tabular datasets to examine obvious patterns and to provide detailed view on specific items at the same time. In our application, pathway nodes at the top level are placed in the center of a cell, edges are routed along the cell borders as intended result of the previously presented layout algorithm. When a node is expanded, the row and the column are enlarged in which the node is situated. Edges leading to and from one of the four ports (see Sec. 4 and Fig. 4 for example) of the pathway node are elongated while the remaining elements keep their relative position.This approach follows Ben Shneiderman's mantra of visual information-seeking: overview first, zoom and filter, details on demand [20]. Our application supports this concept in the following ways:

Overview first. The grid layout algorithm positions top-level nodes on a regular grid where each grid position can be regarded as a cell in a table. The user starts with examining the completely collapsed graph, i.e., only top-level nodes are visible. The application allows to display a node simply by showing the associated pathway's name as caption (see Fig. 2) or by creating an iconized view of the node's nested graph.

Zoom and Filter. We have implemented *semantic zooming* to display labels once a certain threshold is reached. Tool tips add additional information on each pathway node. If enabled, icons in top-level nodes depicting the nested graph give a quick hint on the pathway's size and complexity.

Details on Demand. The user can expand selected pathway nodes to explore the detailed network of chemical reactions. In contrast to the established Table Lens method, an arbitrary number of cells (pathways) can be enlarged (*multiple foci*) and examined in detail (see Fig. 3 and 4). Advanced selection and highlighting techniques facilitate and support the exploration process: selecting a pathway node highlights all objects belonging to that cluster. Selecting a specific reaction node highlights all edges to the associated substrate and product nodes. Selecting a compound node highlights all reactions this compound is involved including its connections to adjacent pathways.

5.2 Design of the Graphical User Interface

The GUI of the visualization software basically consists of three components, see Fig. 6 and 7.

The *Graph View* at the left hand side of the window renders the graph and provides an interface to interact with or edit the topology of the graph directly. Each graphical object can be individually selected, and applicable properties can be assigned via a context menu. The integrated graph editing capability allows

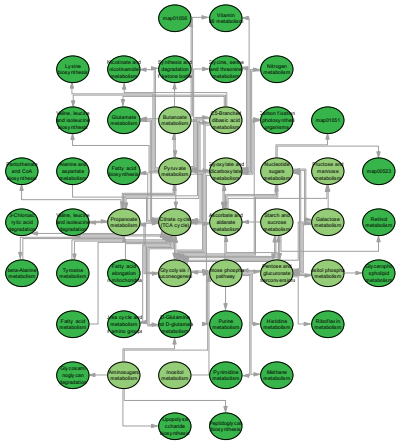

Fig. 2. The top-level graph of the "Carbohydrate Metabolism" (bright) and related pathways (dark). The highlighted nodes can be expanded to reveal the detailed reaction network.

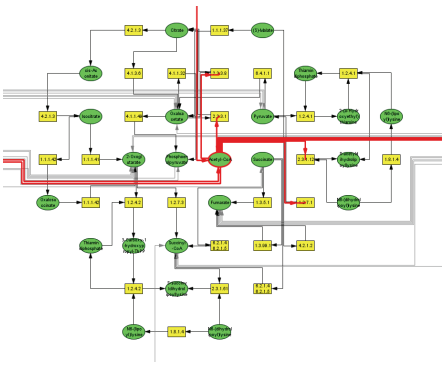

Fig. 3. Detailed view of the expanded node "Citrate Cycle (TCA)". The highlighted compound node "Acetyl-CoA" plays a central role within this pathway and establishes several connections to adjacent pathways.

the user to manually construct pathway graphs or to modify a given layout either generated by the algorithm or loaded from file. Expanding or collapsing individual nodes can be performed by either double-clicking a node or selecting the operation via the associated entry in the context menu.

The *Data Browser* displays the hierarchical structure of the grap (explorer layout) and grants access to textual or numerical attributes of each graph element. Generic graph element properties, e.g., edge width, node size and shape, color or transparency, can be manipulated, and the effects will be directly displayed in the graph view. A simple search function among the textual attributes can be used as a filter to highlight and select a group of graph elements. This is an intuitive way to state queries like *"Select all pathways containing the compound* Pyruvate" (see Fig. 5). Highlighting elements matching a given search pattern is also propagated to the top-level.

The *Algorithm Info Area* at the bottom-right hand side displays textual output giving feedback on the progress of invoked graph or layout algorithms and to present search results, cf. Fig. 7.

6 Performance Results

Our KEGG import routine is suitable to construct pathway graphs of different size and complexity. To implement, test and demonstrate the discussed techniques, we constructed two graphs. Images 2 through 5 were created from 17 pathway files downloaded from the KEGG database covering the complete carbohydrate metabolism. Additional non-expandable pathway nodes were created when referenced in one of the input files. A graph with a total of 649 compound

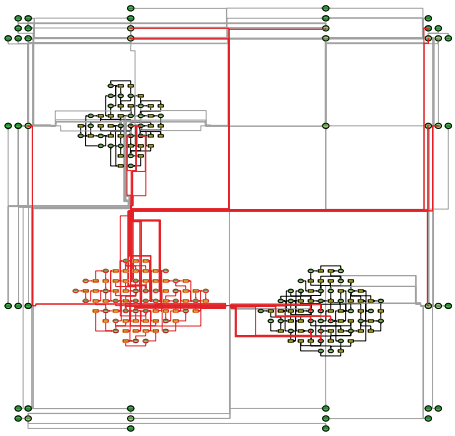

Fig. 4. Three expanded pathway nodes: "Citrate Cycle (TCA)", "Pentose Phosphate Pathway", and "Glycolysis / Gluconeogenesis"; the latter being highlighted in red with the connections to adjacent pathway nodes.

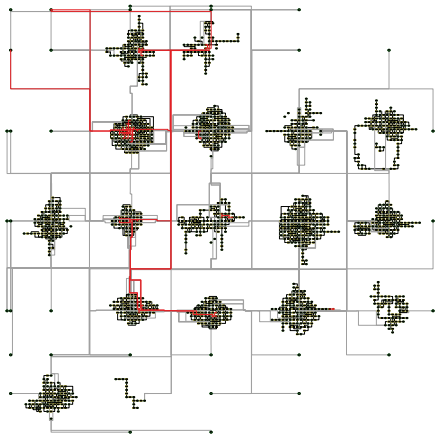

Fig. 5. Bottom-level graph. Reaction network of pathways associated with the carbohydrate metabolism. The search result for "Pyruvate" is highlighted including its incident edges.

nodes, 50 pathway nodes, 814 reaction hyperedges and 149 regular inter-pathway edges was created. The portion of the graph containing the hyperedges and nodes was converted into a bipartite graph, where the previous hyperedges are displayed as rectangular nodes (yellow) labeled with the EC numbers of the catalyzing enzymes and the nodes as ellipses (green) labeled with the compound's chemical name, resulting in a total number of 1,513 nodes and 1,861 edges. This graph could easily be handled by the visualization software. On an Intel® Xeon® (2 GHz, 32 GB RAM) machine, response times of the graphical user interface were less than 0.2 sec for any operation discussed previously enabling a smooth interaction with the displayed graph. A second example was more complex. After converting the hypergraph portion into a bipartite subgraph, the graph describing the complete metabolism of a generalized eucaryotic cell had a total number of 10,067 nodes and 11,706 edges. Depending on the visible portion in the scrollable graph view area, any collapse/expand operation took up to 4 sec if the complete graph was visible, and up to 2 sec if one pathway was located in the visible area. The response times for scrolling, zooming, and highlighting elements for the worst-case scenario (all pathways expanded) were less than 0.75 sec if up to 1/4 of the graph's elements were visible, and less than 0.25 sec if the visible portion was 1/10 of the completely laid out graph.

The runtime of the grid layout algorithm heavily depends on the choice of parameters. For large graphs, the brute-force method testing all grid positions naturally takes longer compared to the simulated annealing method. The choice of the area ratio $a = 4 \cdot |V|$ generally produced more aesthetic layouts for cyclic

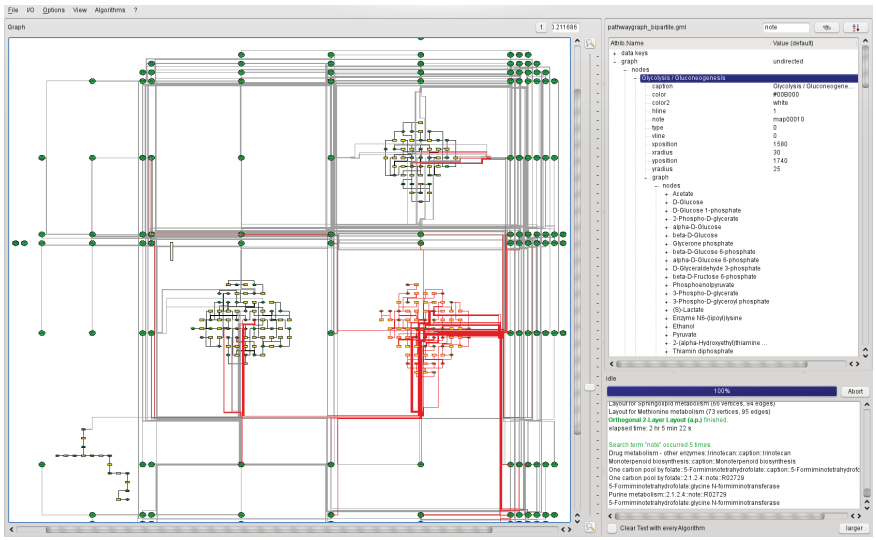

Fig. 6. GUI of the Visualization and Editing Tool. The top-level graph consisting of 154 pathway vertices with 4 expanded pathways. The node "Glycolysis/Gluconeogenesis" was selected in the Data Browser (right, top) resulting in highlighting all its compound and reaction nodes including connections to adjacent pathways.

and chain-like structures because of the larger space available to unfold those substructures, but resulted in increased runtime for the brute-force method.

7 Conclusion

The proposed software is able to layout and display complex graphs with a high number of elements. The development process was intensively accompanied by domain experts from biology and biochemistry. For metabolic pathway networks, not only the graph topology is relevant, a high number of additional attributes—textual annotations in our case—need to be visualized. Semantic zooming and focus&context methods are applied to accomplish this goal, instant highlighting of graph elements fitting the pattern of a string based search operation is an intuitive way to extract specific information on the dataset. The main benefit of the adapted Table Lens method is the preservation of the mental map. Many of the visualization tools lack this key feature. Even though node expansion and collapse produce very discrete and rather abrupt changes in the graph appearance, only the row and the column of the grid position are affected while the remaining elements keep their relative position. In combination with continuous zooming, it is a straightforward task to explore even large graphs. Highlighting individual or groups of edges greatly facilitates the tracking of routes. In the presented grid layout algorithm, vertex placement and edge routing are performed in two separate steps. This offers the opportunity to develop alternative node placement routines fitting the specific needs of pathway visualization in the future.

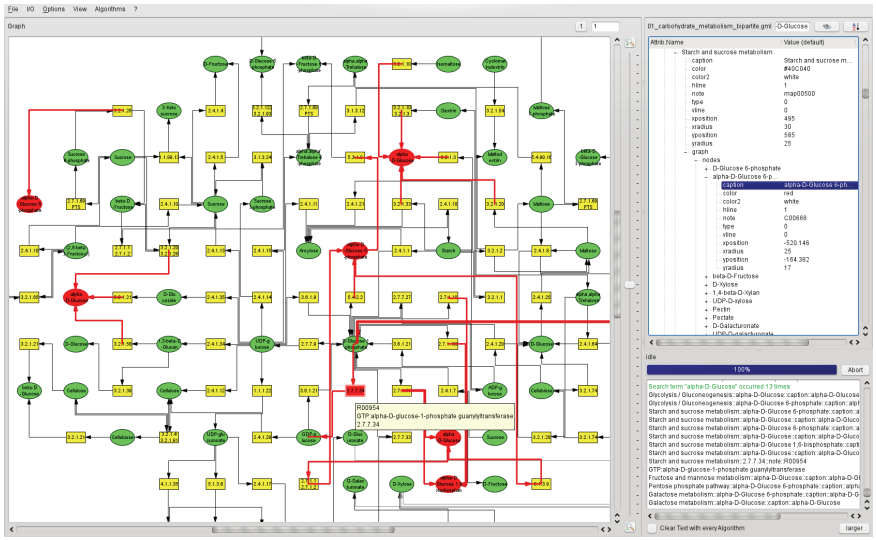

Fig. 7. A more detailed view of the bottom level graph. This portion of the graph displays the pathway *Starch and Sucrose metabolism*. The *Algorithm Info Area* (bottom, right) gives feedback on invoked algorithms and displays search results. In this scenario, a search for the term *alpha-D-Glucose* was performed and resulted in 13 matches being highlighted in the Graph View and in the hierarchical Data Browser view.

Acknowledgments. This work has been funded by the Volkswagen Stiftung under grant I/82 719.

References

1. Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout Adjustment and the Mental Map. Journal of Visual Languages and Computing 6 (2), 183–210 (1995)
2. Rao, R., Card, S.K.: The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 318–322. ACM, New York (1994)
3. Karp, P.D., Paley, S.M., Romero, P.: The Pathway Tools software. Bioinformatics 18, 225–232 (2002)
4. Hu, Z., Mellor, J., Wu, J., DeLisi, C.: VisANT: an online visualization and analysis tool for biological interaction data. BMC Bioinformatics 5 (1), e17 (2004)
5. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Gen. Res. 13 (11), 2498–2504 (2003)
6. Schreiber, F.: High Quality Visualization of Biochemical Pathways in BioPath. In Silico Biology 2 (2), 59–73 (2002)
7. Karp, P.D., Ouzounis, C.A., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahren, D., Tsoka, S., Darzentas, N., Kunin, V., Lopez-Bigas, N.: Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Research 33 (19), 6083–6089 (2005)

8. DiBattista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall, New Jersey (1999)
9. Nicholson, D.E.: Metabolic Pathways Map (Poster). Sigma Chemical Co., St. Louis (1997)
10. Lungu, M., Xu, K.: Biomedical Information Visualization. In: Human-Centered Visualization Environments, pp. 311–342., Springer (2006)
11. Kerren, A., Ebert, A., Meyer, J.: Human-Centered Visualization Environments, GI-Dagstuhl Research Seminar. In: LNCS, vol. 4417, Springer (2007)
12. Kerren, A.: Interactive Visualization and Automatic Analysis of Metabolic Networks – A Project Idea. Technical Report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria (2003)
13. Albrecht, M., Kerren, A., Klein, K., Kohlbacher, O., Mutzel, P., Paul, W., Schreiber, F., Wybrow, M.: On Open Problems in Biological Network Visualization. In: Proc. of the 17th International Symposium on Graph Drawing (to appear), Chicago, USA, Springer (2009)
14. Albrecht, M., Kerren, A., Klein, K., Kohlbacher, O., Mutzel, P., Paul, W., Schreiber, F., Wybrow, M.: A Graph-drawing Perspective to Some Open Problems in Molecular Biology. Technical report TR08-01-003, Lehrstuhl XI für Algorithm Engineering, Fakultät für Informatik, TU Dortmund, Germany (2008)
15. Brandes, U., Dwyer, T., Schreiber, F.: Visualizing Related Metabolic Pathways in Two and a Half Dimensions. In: Proc. International Symposium on Graph Drawing (GD'03), pp. 111–122, LNCS, vol. 2912 (2003)
16. Kyoto Encyclopedia of Genes and Genomes, `http://www.kegg.jp/kegg/`
17. Klukas, C., Schreiber, F.: Dynamic exploration and editing of KEGG pathway diagrams. Bioinformatics 23 (3), 344–350 (2007)
18. Michal, G.: Biochemical Pathways: Biochemie-Atlas. Spektrum Akademischer Verlag, Heidelberg (1999)
19. Saraiya, P., North, C., Duca, K.: Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. Information Visualization 4 (3), 191–205 (2005)
20. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: VL, pp. 336–343 (1996)
21. Streit, M., Kalkusch, M., Kashofer, K., Schmalstieg, D.: Navigation and Exploration of Interconnected Pathways. Eurographics / IEEE-VGTC Symposium on Visualization 27 (3) (2008)
22. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Inf. Process. Lett. 31 (1), 7–15 (1989)
23. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
24. Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 4 (2), 100–107 (1968)
25. Batini, c., Nardelli, E., Tamassia, R.: A Layout Algorithm for Data Flow Diagrams. IEEE Trans. Software Eng. 12 (4), 538–546 (1986)
26. Kaufmann, M., Wagner, D.: Drawing Graphs, Methods and Models. In: Drawing Graphs, LNCS, vol. 2025, Springer (2001)