Visualization of Metabolic Networks

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet

INFORMATIK

Vorgelegt von

Dipl.-Inf. Markus Rohrschneider

geboren am 14.08.1977 in Leipzig

Die Annahme der Dissertation wurde empfohlen von:
1. Prof. Dr. Gerik Scheuermann, Universität Leipzig
2. Prof. Dr. Falk Schreiber, Monash University

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteidung am 26.01.2015 mit dem Gesamtprädikat

Magna cum laude

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Markus Rohrschneider

Leipzig, 31. Januar 2015

Acknowledgement

This contribution would not have been possible without the support of my coworkers of the Signal and Image Processing group and the Bioinformatics department, as well as the encouragement of my friends and my family.

More specifically, I want to thank Andreas Kerren for the help on the related works chapter, Christian Heine and André Reichenbach for implementing the grid layout algorithm. Christian also greatly contributed to chapter 3.2 and provided the *libgraph* library, which supplies the core data structure for *GraphEdit*.

I also want to thank Alexander Ullrich for the fruitful cooperation in the visualization of dynamic graphs and the supply of simulation data representing evolving metabolic networks.

Furthermore, I would like to thank Peter Stadler for sharing his expert knowlege in the fields of biology and chemistry and for providing the bioinformatical background in section 5.2.

Last not least I thank Gerik Scheuermann for supervising my work, his insightful contributions, continuous enthusiasm and motivational comments.

Abstract

The metabolism constitutes the universe of biochemical reactions taking place in a cell of an organism. These processes include the synthesis, transformation, and degradation of molecules for an organism to grow, to reproduce and to interact with its environment. A good way to capture the complexity of these processes is the representation as metabolic network, in which sets of molecules are transformed into products by a chemical reaction, and the products are being processed further. The underlying graph model allows a structural analysis of this network using established graphtheoretical algorithms on the one hand, and a visual representation by applying layout algorithms combined with information visualization techniques on the other.

In this thesis we will take a look at three different aspects of graph visualization within the context of biochemical systems: the representation and interactive exploration of static networks, the visual analysis of dynamic networks, and the comparison of two network graphs. We will demonstrate, how established infovis techniques can be combined with new algorithms and applied to specific problems in the area of metabolic network visualization.

We reconstruct the metabolic network covering the complete set of chemical reactions present in a generalized eucaryotic cell from real world data available from a popular metabolic pathway data base and present a suitable data structure. As the constructed network is very large, it is not feasible for the display as a whole. Instead, we introduce a technique to analyse this static network in a top-down approach starting with an overview and displaying detailed reaction networks on demand. This exploration method is also applied to compare metabolic networks in different species and from different resources. As for the analysis of dynamic networks, we present a framework to capture changes in the connectivity as well as changes in the attributes associated with the network's elements.

Contents

1 Introduction 7
1.1 Definitions . . . 8
1.2 Metabolic Networks as Hierarchical Hypergraphs . . . 10

2 Related Work 12

3 Focus&Context View for Static Metabolic Networks 19
3.1 Network Data Source . . . 20
3.1.1 KEGG Database . . . 20
3.1.2 Data Acquisition and Graph Construction . . . 21
3.2 Hierarchical Graph Layout . . . 24
3.2.1 Vertex Position . . . 25
3.2.2 Edge Routing . . . 28
3.2.3 Edge Bundling . . . 29
3.2.4 Alternative Grid Generation . . . 30
3.3 Graph Exploration . . . 33
3.3.1 Hierarchical Graph Navigation . . . 33
3.3.2 Semantic Zoom . . . 34
3.3.3 Selection and Highlighting . . . 38
3.4 Performance Results . . . 39
3.5 Conclusion . . . 42

4 Visualization of Dynamic Graphs 44
4.1 Evolution of Metabolic Networks . . . 44
4.2 The Model . . . 46
4.3 The Data . . . 48
4.4 Visual Data Exploration . . . 49
4.4.1 Overview . . . 51
4.4.2 Zoom and Filter . . . 53
4.4.3 Detailed View . . . 56
4.5 Results . . . 56
4.6 Conclusion . . . 62

5 Comparative Visualization of Network Graphs **63**
5.1 Inter-species Comparison of KEGG Pathway Networks 64
5.1.1 Organism Overview Visualization 65
5.1.2 Chemical Network Comparison 67
5.1.3 Results 68
5.2 Visual Comparison of KEGG Pathway and BioCyc content . . 71
5.2.1 Motivation 71
5.2.2 Data Resources, Preprocessing, and Data Structure 73
5.2.3 Graph Matching 74
5.2.4 Visual Comparison and Exploration 75
5.2.5 Results 80
5.2.6 Concluding remarks 85

6 Visualization Framework **87**
6.1 Graphical User Interface 87
6.2 Algorithms in GraphEdit 89

7 Conclusion **90**

8 Appendix **92**

References **96**

Curriculum Vitae **103**

List of publications **104**

1 Introduction

Graph visualization plays a central role in the field of information visualization. This is especially true for applications in the life sciences, as many biological systems can be modelled as networks. Examples are *regulatory networks* describing complex interaction patterns of chemical compounds with DNA to control gene expression and protein synthesis. In *signaling pathways* chemical signals invoke cellular reactions by interacting with receptor proteins triggering a cascade of intracellular processes. A *Metabolic network* represents the set of chemical reactions transforming molecules. The interactions and relations between different chemical molecules – or compounds – define a highly complex network augmented with domain-specific annotations. These annotations provide semantic information to the network and may include molecule structures, chemical names of compounds, enzymes catalyzing certain chemical reactions, reaction kinetics, or concentration values.

In biochemistry, these networks are often clustered into so-called metabolic pathways. The term pathway in this context refers to a subset of this network performing a specific biological function, such as the synthesis, transformation, or degradation of certain organic substances in biological systems. The classification is inferred from expert knowledge and takes into account functional properties of the subsets as well as spatial characteristics. Often the location of specific substances within a biological cell is limited to specialized cell compartments providing more criteria to distinguish between different pathways.

This thesis will demonstrate how graph visualization techniques can be applied to common problems in the work with chemical networks in bioinformatics research. In the following, we give some formal definitions of graph-related terms used in this thesis. Chapter 2 will give an overview on related work and applications designed for the work with complex networks in general and metabolic networks in particular.

Chapters 3 through 5 address three major tasks in the domain of metabolic network visualization: representation and navigation techniques for static networks, the exploration of dynamic metabolic pathways, and the comparison of chemical networks based on two scenarios – the metabolism in different species and the comparison of metabolic network data retrieved from two major bioinformatics resources.

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. The software mentioned throughout this work was developed as part of the project. A detailed description of the framework is given in chapter 6.

1.1 Definitions

A (simple) *graph* $G = (V, E)$ consists of a finite set of vertices V and a set of edges $E \subseteq \{(u, v)|u, v \in V, u \neq v\}$. Each edge $e = (u, v)$ is *incident* to the vertices u and v. Both u and v are then called *adjacent*. A *digraph* $G = (V, E)$ is a graph with oriented edges called *arcs*. A *path* is a sequence of unique edges that connects vertices u and v. A *cycle* is a non-empty path from u to u. A graph is *connected* if for each pair of vertices u, v there exists a path from u to v. A graph or digraph is *acyclic* if it contains no cycle. A *directed acyclic graph* or *DAG* is an acyclic digraph.

A *hypergraph* $H = (V, E)$ is an extension of a graph allowing *hyperedges* of E to connect multiple vertices: $e = (V_{in}, V_{out})$ with $V_{in}, V_{out} \subset V$ and $V_{in} \neq \emptyset$, $V_{out} \neq \emptyset$. 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 $u \in V_{in}$ or product $v \in V_{out}$ as shown in Fig. 1.1. In the hypergraph model, a *regular edge* is a hyperedge with $\mid V_{in} \mid= 1$ and $\mid V_{out} \mid= 1$. We use the term *regular edge* to represent relations between nodes other than chemical reactions. A *half-edge* establishes the link between a vertex $u \in V_{in}$ or a vertex $v \in V_{out}$ and the edge e. We say that $e = (V_{in}, V_{out})$ has $\mid V_{in} \mid$ *incoming* and $\mid V_{out} \mid$ *outgoing* half-edges.

A *tree* is a connected graph with n nodes and $(n - 1)$ edges not containing any cycles. The height is the length of the longest path in the tree. There exists exactly one path between any two nodes. In directed trees, there exists exactly one node without incoming edges called *root*. All nodes without outgoing edges are called *leaves*. If a graph is not connected and each connected component is a tree, we call this graph a *forest*.

A *bipartite graph* $G_{r,c} = (V_r, V_c, E)$ is a (simple) graph with two types of vertices V_r and V_c ($V_r \cap V_c = \emptyset$). Edges in a bipartite graph may only connect nodes of different types, i.e., $E \subseteq \{(u,v) | u \in V_r, v \in V_c \vee u \in V_c, v \in V_r\}$. It is often desirable to work with bipartite graphs instead of hypergraphs, since common graph algorithms for layout or traversal, for example, are defined on (simple) graphs rather than on hypergraphs. Bipartite graphs represent hypergraphs without loss of information and can be constructed simply by adding vertices to V_r for every hyperedge $e \in E$. The original (compound) nodes are the vertices V_c. For every hyperedge $e = (V_{in}, V_{out})$ we add edges to the bipartite graph $G_{r,c}$ representing the half-edges in H: For every incoming half-edge $(u, e), u \in V_{in}$ we add an edge $(u_1, u_2), u_1 \in V_c, u_2 \in V_r$, and for every outgoing half-edge $(e, v), v \in V_{out}$ we add an edge $(v_1, v_2), v_1 \in V_r, v_2 \in V_c$. In the bipartite version of the graph, V_r represents the set of chemical reactions, and V_c represents compounds.

The *drawing* or *layout* of a graph assigns vertices to points and edges to curves in the image space, which is usually the 2D drawing plane. In some definitions, the size of the node is included in the layout description.

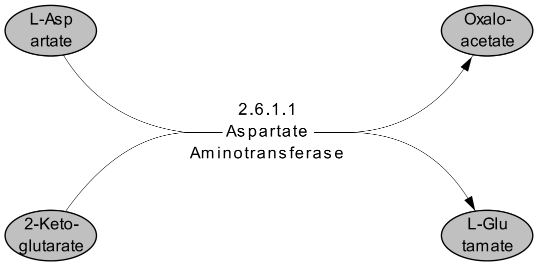


Figure 1.1: A hyperedge which represents a biochemical reaction. The direction of the hyperedge specifies, whether a compound is substrate – *L-Aspartate* and *2-Ketoglutarate* – or product – *Oxaloacetate* and *L-Glutamate*. Note that many reactions are reversible. In those cases, the direction of the hyperedge simply gives a hint on the reaction's chemical equilibrium.

1.2 Metabolic Networks as Hierarchical Hypergraphs

Based on the clustering of the metabolic network into pathways, we can define a hierarchy on the graph. This hierarchy can be represented by an additional (simple) graph. The graph is either a DAG with exactly two layers for overlapping subsets or a forest with trees of height 1 for non-overlapping subsets of the chemical network. In a reaction network with unified compound nodes, i.e., there is exactly one node representing a specific chemical species, the clustering into pathways therefore results in an overlapping network. The intersection of two clusters can be considered as biomass exchange between the two pathways. We call two pathways with a non-empty intersection of their compound node sets as *adjacent*. To convert an overlapping hierarchy into a non-overlapping, compound nodes being present in more than one pathway are duplicated and appear in each respective pathway. Within a pathway, reaction and compound nodes are always unified.

To obtain a hierarchical graph, each metabolic pathway is represented by a node at the top level. In the case of non-overlapping pathways, these nodes represent the root nodes in the forest. The set of compound nodes constitutes the bottom-level of the hierarchy graph.

We augment the hierarchy graph with different sets of edges:

1. For each reaction in the metabolic network, we add a hyperedge connecting the substrate nodes with the product nodes of the respective chemical reaction.
2. In non-overlapping pathway graphs, the biomass exchange between two adjacent pathways is indicated by a regular edge connecting two nodes representing the same chemical species in different pathways.
3. The connection indicating a biomass exchange is propagated to the top-level to show that two pathways are adjacent. In section 3.1.2 we introduce the concept of *virtual edges* supporting node expansion and collapse to navigate the hierarchy. Virtual edges are (regular) edges representing adjacency relations between different clusters.

Fig.1.2 depicts three metabolic pathways represented as a non-overlapping hierarchy. In practice, non-overlapping hierarchical graphs are easier to handle with respect to the layout computation and exploratory interaction. In a hierarchical

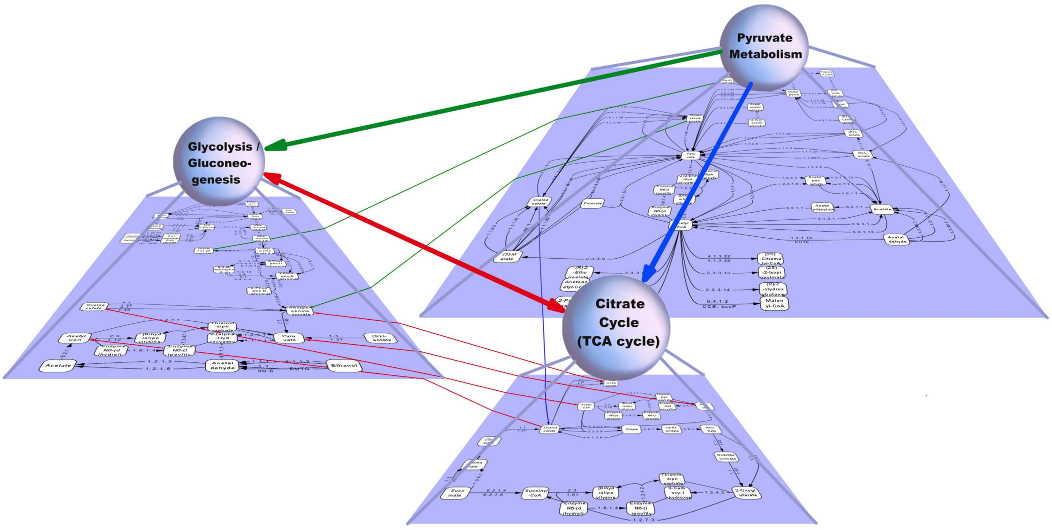


Figure 1.2: 2-layer hierarchy of a non-overlapping pathway graph. Each top-level node represents a pathway, which has the biochemical network as nested graph. The bottom-level nodes represent chemical compounds, hyperedges represent chemical reactions. To obtain a forest-like hierarchy, elements present in more than one pathway will be duplicated and appear in a pathway at most once. Links between pathways are established by regular edges between identical chemical compounds located in different pathways to indicate a flux from one pathway to another. These links are propagated to the top-level reflecting pathway adjacencies.

graph represented by a forest-like hierarchy, each cluster can be layouted independently, 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 [4]. Further important issues are the preservation of the so-called mental map [51] 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) [43]. 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 [15].

Cytoscape [68] 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 the network complexity. For instance, the user can select nodes and edges according to their name and other attributes. This system also allows a simple mapping of data attributes to visual elements (mainly color and labels) of nodes and edges. VisANT [34] 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, for example as scatter plots.

Especially for metabolic networks, large and hand-drawn posters were produced in the past, for example, Nicholson's pathway map [54] or the widely-used metabolic pathway poster published by Roche Applied Science [50]. Other projects have created graphical representations of metabolic networks and offer them via web pages (e.g., the BioCyc collection [40]). The widespread pathway drawings of the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [42], see Section 3.1.1, 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 [4]. A set of visualization tools for metabolic networks is available that follow these established drawing styles. We will briefly describe a choice of related work in the following.

The software BioPath [66] is a web-based visualization tool for metabolic pathways that also refers to the aforementioned posters by taking over its drawing conventions and data. The developers propose a number of requirements for the visualization of biochemical networks to make the resulting diagrams acceptable for the domain experts. The drawings are produced by a Sugiyama-style hierarchical layout algorithm allowing restrictions on the horizontal and vertical order of nodes both for preserving the mental map and for drawing cyclic subnetworks in a special way. Thus, they respect the conventions commonly used in the hand-made diagrams of the biologists. However, there are no means of influencing the drawings interactively. This drawback was partly resolved by the KGML-ED system of the same authors [46] through the possibility to delete or add network elements or to change labels and colors.

Another method for drawing metabolic networks automatically was devised by Becker and Rojas [6]. For several kinds of subnetworks, different layout algorithms are used (hierarchical or force-directed). The subnetwork drawings are then combined to a global drawing. This idea has a positive effect on the layouts, but the partition is only obtained heuristically. However, the good readability of the drawings is also due to the fact that neither side metabolites nor enzymes are displayed. No interaction is possible, and neither clustering nor preservation of mental maps is implemented.

Other tools produce better layouts but do not allow to group, focus on, fix, or even format parts of the graph, e.g., PathFinder [27]. This tool allows to identify paths between metabolites and to compare reaction paths with own data. The Pathway tools [41] stand out due to their ability to draw metabolic networks in different steps of abstraction, which is obtained by using symbols for certain subgraphs. This approach could lead to a better readability in general; the quality of the drawings, however, needs to be improved.

PathBank [32] is a web-based 3D pathway visualization system with a database back end that integrates data from various sources. Pathways can be automatically visualized in two or three dimensions. This is coupled with standard interactions that allow the user to zoom, pan, rotate, and drag. A number of layout algorithms are available, but they are mainly based on traditional methods that do not consider pathway conventions.

A further 3D approach was presented by Rojdestvenski. His Metabolic Network Visualizer [63] draws metabolic networks in three dimensions in order to avoid crossings between edges. Interaction is again limited to standard techniques for navigation. Other disadvantages are the unusual visual representation that will hardly be accepted by the user and the inadequate visualization of larger networks.

In summary, none of the above discussed tools fully meets the requirements for the visualization of metabolic networks. Some approaches already fail when producing static drawings, either because they were developed by computer scientists using standard graph drawing techniques without adapting them to the special situation, or because they were developed by domain experts who are not familiar with visualization techniques. Additionally, most tools lack interaction techniques, such as focus&context, and does not scale for the same reason. A visual representation of results from abstract analyses, experimental series, or simulations is usually impossible or in its infancy [4].

Newer approaches are based on a close interdisciplinary work between researchers in visualization and biochemistry. An example is the Caleydo framework [71] that extends the standard pathways of KEGG into 2.5D, similar to the report of Kerren [43] and the work of Brandes et al. [8], 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. [64] discussed in their paper 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 [49]):

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.

Six pathway visualization systems were evaluated according to these requirements. As result, biologists are usually reluctant to use these systems, because of the absence of the previously listed requirements. Also, they prefer visualizations that are similar to those in text books they are working with.

Currently, our approach to visualize static metabolic networks as described in chapter 3 addresses several of the aforementioned requirements and improves the most previous work by using 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 eukaryotic cell (Req. 4) by following Shneiderman's mantra [69]: overview first, zoom and filter, details on demand. If the user explores the pathways interactively, the visualization approach preserves the mental map. 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. The integration of more complicated attributes as well as biological patterns regarding topological substructures are still missing. Here, other tools, such as BioPath, still have an advantage to be fully accepted by biologists.

A good introduction from a general point of view on the visualization of dynamic – by the authors also referred to as longitudinal – networks is given in the article Bender-DeMoll and McFarland [7]. The authors propose a framework for the visualization of social networks and their dynamics and present the tool *SoNIA* (Social Network Image Animator, `http://sonia.stanford.edu`) in which they compare different layouts and navigation techniques. Their work addresses general issues in the field of dynamic network visualization with applications to social interactions and refers topological changes, i.e., dynamics in the network's connectivity, only. In biological systems, however, much information is also stored in the set of attributes associated with the network elements, that is subject to change over time as well, which will be discussed in chapter 4. An overview on open problems and challenges in biological network visualization is provided by the papers [62, 5]. They provide a comprehensive list of related work, however not focused on the visualization of dynamic biochemical pathways. Oldiges et al. [56] address the specific problem of metabolic network model visualization. Nevertheless, their article is particularly related to the numerical analysis of dynamic biochemical systems with less emphasis on the visual analysis of the dynamics of the network topology.

In general, the visualization of dynamic graphs is a well-known area in the graph drawing community [10]. Dynamic graph drawing addresses the problem to layout graphs, which evolve over time by adding and deleting edges and nodes. This results in an additional esthetic criterion known as *preserving the mental map* [51]. Ad-hoc approaches compute a new layout for the entire graph after each time step using algorithms developed for static graph layout, see for example those presented in the book [15]. In most cases this approach produces layouts which violate the mental map. One solution of this problem is to apply a technique known from key-frame animations called *inbetweening* to achieve smooth transitions between subsequent graphs, i.e., animations show how nodes are moved to their new positions. Prominent examples were presented by various authors [19, 48, 20, 47]. In our own work, we follow the basic idea of the so-called *Foresighted Layout* (FL) of dynamic graphs [17]. Given a sequence of n graphs, a global layout is computed, which induces a layout for each of the n graphs. The FL-algorithm is generic in the sense that it takes a static graph drawing algorithm as a parameter. It optimally preserves the mental map, but this can lead to an oddly visual appearance at the beginning of the resulting graph animation. A subsequent extension of the original approach improved this drawback [16]. An algorithm for drawing a sequence of graphs online, i.e., where the graph sequence to be laid out is not known in advance, was presented by Frishman and Tal [23].

The general design of our plug-in as described in chapter 4 is based on standard coordinated and multiple view visualization techniques. An excellent starting point for related work of this kind of visualization techniques is the annual conference series on Coordinated & Multiple Views in Exploratory Visualization (CMV) or the work of Roberts [61]. In our case, the coordination between the different views is mainly done by brushing techniques. The work of Moody et al. [52] focuses on the visualization of dynamic networks in general and the evolution of social networks in particular. The authors state two common approaches: plotting network summary statistics as line graphs over time and examining separate images of the network at each point in time. Our work has been inspired by these two techniques.

Some important publications on the specific topic of metabolic network comparison can be found in [67] and [3]. Albrecht et al. focus on finding a suitable layout for the union graph, which is constructed from the two individual graphs to be compared. The union graph layout is used to layout common subsets, while still preserving the differences. The algorithm is designed in a hierarchic manner. An overview graph construction is provided by laying out the backbone first, and the detailed graph representing changes is constructed and laid out afterwards. While the work in [3] addresses generic biological networks, a comparison of similar metabolic pathway graphs is given in [67]. Identical parts of the network are identified to define constraints for a common layout. The proposed method is applied to data obtained from the BioPath System and the KEGG Pathway database. A different, non-visual approach of comparing metabolic networks is found in [79]. A cross-species comparison (archae, bacteria, eukaryotes) is based on topological properties, i.e., network indices, degree distribution measures, and motive profile measures. For the visualization of these networks, the algorithm of Kamada and Kawai [36] algorithm was used to optimize the layout.

The two bioinformatics resources we use in this work provide very limited capabilities for comparing different metabolic networks. The KEGG web interface allows the user to project metabolic pathways specific to one of approximately

1400 organisms onto the reference pathway diagrams. The respective enzymes are highlighted. The web interface of BioCyc [12] is more flexible in that respect. Pathways are dynamically rendered with a user-specified level of detail. A cross-species comparison is possible for two organisms. MetaCyc also provides the software package *Pathway Tools* [41] for navigating the data, metabolic pathway prediction, analysis and visualization.

3 Focus&Context View for Static Metabolic Networks

In this chapter, we introduce methods for the exploration of a metabolic network. When investigating those networks, we can identify three different scales – or levels of detail – being of interest for the researcher:

1. Pathway graph (top-level) consisting of functional units and their relations to each other in terms of biomass transfer.
2. Reaction graph (bottom-level) containing nodes and edges representing the transformation of chemical compounds.
3. Molecular level describing the chemical structure of compounds.

The objective is to integrate all three levels into one framework allowing a seemless transition between the scales. This is crucial for the preservation of the mental map when navigating the network.

We take the hierarchical structure of the graph modelling the network into account for the layout computation and exploit the hierarchy for navigation within the data. The exploration process is performed in a top-down manner following Ben Shneiderman's mantra of information visualization: overview first, zoom and filter, details on demand. Two interaction techniques are combined for that purpose: expansion of top-level pathway nodes to reveal the detailed reaction network of pathways of interest, and a semantic zoom displaying more and more properties associated with the nodes of the graph as drawing space becomes available.

To support the domain expert in the exploration and analysis process, we adapt the well-known Table Lens [60] metaphor with the possibility to select multiple foci. Here, we regard each grid position in our layout as a cell in a table that can be expanded to locally reveal lower levels in the hierarchy. The layout places the network nodes on a fixed rectilinear grid and routes the edges orthogonally between the node positions. The 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.

Furthermore, our visualization tool offers additional features, such as highlighting of individual paths. We successfully applied our method to provide interactive access to the complete biochemical network stored in the KEGG database.

In the following section, we present the graph generation from metabolic network data provided by KEGG PATHWAY to generate a reasonably realistic scenario for testing and demonstrating the implemented methods. Section 3.2 describes the layout algorithm used as basis for the interaction and navigation techniques discussed in Section 3.3. The results and concluding remarks are presented in Sections 3.4 and 3.5.

3.1 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) System [42] provides annotated pathway data facilitating the construction of metabolic pathway graphs of different sizes. In the following section we give a short summary on the contents and accessibility of the KEGG System, and how the used example graphs were generated.

3.1.1 KEGG Database

Kyoto Encyclopedia of Genes and Genomes 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 live sciences. KEGG consists of 19 individual databases establishing three major components [37].

Systems information: KEGG PATHWAY represents higher order functions presented as a network of interacting molecules. It contains graphical diagrams of cellular processes, such as metabolic pathways, regulatory networks, membrane transport, and cell cycle. Additionally, pathway modules are stored in KEGG MODULE; and KEGG DISEASE represents molecular-level knowledge on diseases. BRITE supplements KEGG PATHWAY and the genomic resources representing functional hierarchies of biological systems. It is a collection of classifications based on an object's biochemical role. This part of the KEGG database incorporates many types of relationships not limited to molecular interactions and reactions.

Genomic information: it is gathered in nine databases to present gene catalogs of all organisms with completely sequenced genomes and selected organisms with partial genome sequences [38, 55]. Comprehensive annotations on genomes and chemical compounds are included in the KEGG ORTHOLOGY system.

Chemical information: KEGG LIGAND is a union of six databases comprising the knowledge on chemical compounds, particularly metabolites, drugs, glycans, enzymes, and enzymatic reactions.

3.1.2 Data Acquisition and Graph Construction

The KEGG System provides several ways to access its contents. Via a web interface the user can examine single pathways displayed as manually drawn images (Figure 3.1). Each node in the illustration is linked to a database entry of the associated compound or enzyme revealing information on the structure, chemical properties and participation in other pathways. Pathway maps are also linked to adjacent pathways indicating that a compound is shared by both pathways, i.e., a biomass exchange from one pathway to another.

Pathway data can be downloaded from a daily updated ftp archive. Pathway maps are stored as XML representation in KGML (KEGG Markup Language) [44] files. KGML is an exchange format intended to describe KEGG graph objects. Each KGML file contains entry elements for three types of graph objects: *boxes* either represent maps of adjacent pathways or enzymes with a reference to a reaction element, *circles* represent compounds. Relation elements may have two types

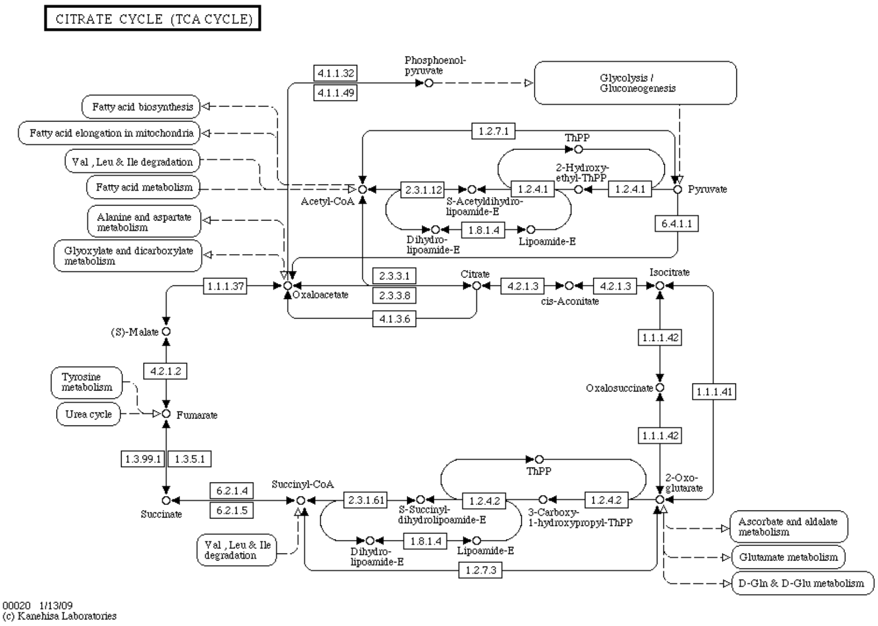


Figure 3.1: Illustration of the central pathway *Citrate cycle* taken from KEGG.

within a metabolic pathway KGML file: ECrel for intra-pathway links to create connections between a pair of enzyme nodes and a compound node in between the two enzymes, and maplink for inter-pathway links to create connections between a compound node and an adjacent pathway.

To develop our novel layout algorithm and implement suitable interaction and navigation techniques, we used KEGG PATHWAY as primary data source to create a realistic scenario. Constructing a hierarchical directed hypergraph with two layers involves four steps:

1. Add a node to the hypergraph for each KGML file specifying a pathway. For each enzyme entry found in the file, add an empty hyperedge (not connected to any node), and for each compound entry add a node to the pathway node's children.
2. Add half-edges to link the hyperedges with in- and out-nodes within a pathway by evaluating the KGML file's relation elements of type ECrel. The direction of the hyperedge is determined by the associated reaction element,

which specifies two sets of compounds (substrates and products).

3. Evaluate the maplink relations to connect identical compounds occurring in the current and the referenced pathway via a regular edge, i.e., a hyperedge having input and output sets with exactly one element. Regular edges simply express links between pathways realized by substances shared by two pathway graphs, and if directed, a flux from one pathway to another. If the referenced pathway was not loaded from a KGML file in (1), simply add a (non-expandable) node and connect it with the compound node of the current pathway specified in the *"maplink"*.
4. Add *virtual edges* for each edge connecting two identical compound nodes in different pathways (see below).

The evaluation of KGML files and converting the data into a graph is similarly done and discussed in [46] in more detail. The result of the KGML conversion of a single pathway is shown in Figure 3.2. We used our visualization tool to display the graph and to manually route the edges for improved clarity. Node positions were taken directly from the KGML description.

To construct a hierarchical graph supporting interactive operations such as node expansion and collapse, we follow the concept of *virtual edges* as depicted in Figure 3.3. When a bottom-level subgraph belonging to a single cluster is collapsed, all elements of the subgraph are hidden and the elements' parent node is displayed instead. Obviously, edges incident on invisible nodes will also be hidden. To propagate connections through all hierarchy levels regardless the expansion state of the graph, edges connecting sublevel nodes are replaced by their virtual edges. In a hierarchical graph with two layers, each inter-pathway edge connecting two bottom-level nodes having different parents requires the construction of up to three virtual edges: one edge connecting the two parent nodes, i.e., pathway nodes, and two edges symmetrically connecting the compound node on the bottom level with the opposite compound's parent node. To avoid a large number of multi-edges in the graph, a virtual edge is inserted only once and a counter keeps track of the number of bottom-level edges it represents.

In addition to the topology of the graph, we store a number of textual attributes to each graph element to provide that information on demand.

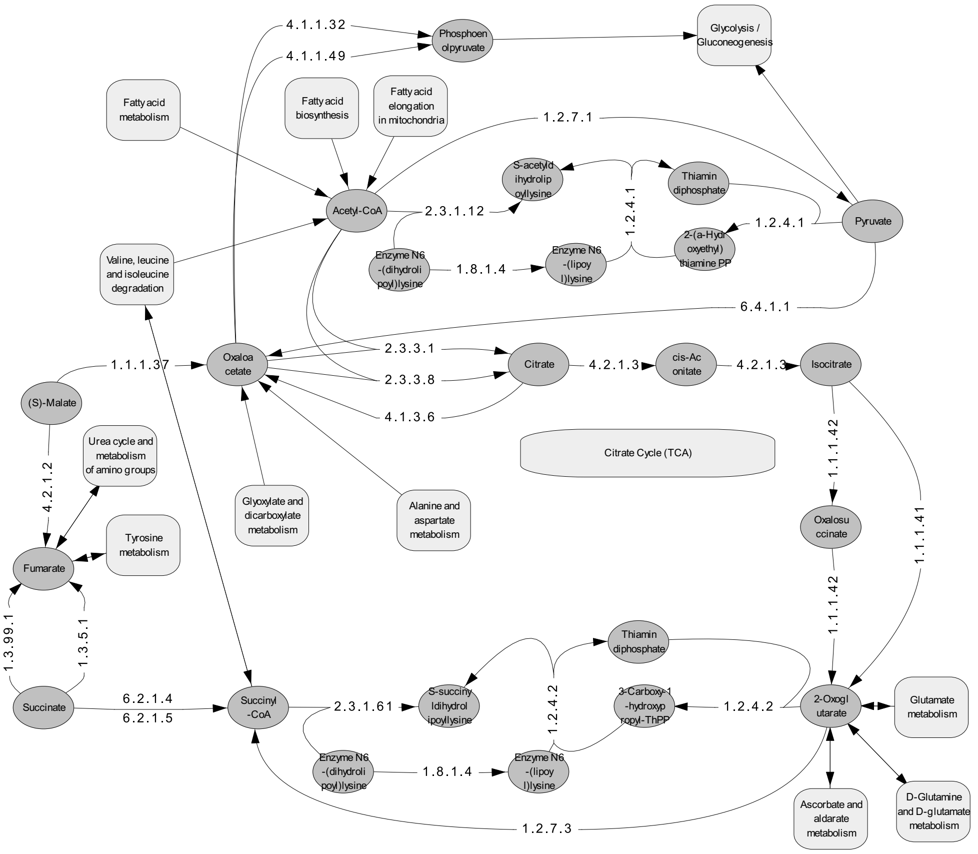

4.1.1.32
4.1.1.49
Phosphoen olpyruvate
Glycolysis / Gluconeogenesis
Fatty acid metabolism
Fatty acid biosynthesis
Fatty acid elongation in mitochondria
1.2.7.1
S-acetyld ihydrolip oyllysine
Thiamin diphosphate
1.2.4.1
Acetyl-CoA
2.3.1.12
1.2.4.1
Pyruvate
2-(a-Hydr oxyethyl) thiamine PP
Enzyme N6 -(dihydroli poyl)lysine
1.8.1.4
Enzyme N6 -(lipoy l)lysine
Valine, leucine and isoleucine degradation
6.4.1.1
2.3.3.1
1.1.1.37
Oxaloa cetate
Citrate
4.2.1.3
cis-Ac onitate
4.2.1.3
Isocitrate
2.3.3.8
(S)-Malate
4.1.3.6
Citrate Cycle (TCA)
Urea cycle and metabolism of amino groups
1.1.1.42
4.2.1.2
Glyoxylate and dicarboxylate metabolism
Alanine and aspartate metabolism
Oxalosu ccinate
1.1.1.41
Tyrosine metabolism
Fumarate
1.1.1.42
1.3.99.1
1.3.5.1
Thiamin diphosphate
Succinate
6.2.1.4
6.2.1.5
Succinyl -CoA
2.3.1.61
S-succiny ldihydrol ipoyllysine
1.2.4.2
3-Carboxy-1 -hydroxyp ropyl-ThPP
1.2.4.2
2-Oxogl utarate
Glutamate metabolism
Enzyme N6 -(dihydroli poyl)lysine
1.8.1.4
Enzyme N6 -(lipoy l)lysine
Ascorbate and aldarate metabolism
D-Glutamine and D-glutamate metabolism
1.2.7.3

Figure 3.2: Conversion from the KGML description to a hypergraph and rendered using our graph visualization software. Node positions were taken directly from the graph element description, edges were manually routed.

3.2 Hierarchical Graph Layout

We model the data in the KEGG database as a hierarchical hypergraph. Each pathway is a graph in the hierarchy. If two pathways exchange compounds, there are maplinks connecting both pathways, and we add regular edges in the top level graph consisting of all pathways as vertices.

The layout of the hierarchical KEGG hypergraph is generated by laying out the corresponding hierarchical *regular* graph. We consider a graph as regular, if it only contains regular edges connecting exactly two vertices. The top level graph is already a regular graph and each pathway hypergraph is converted to its corresponding bipartite graph consisting of compounds and reactions as vertices and an edge exists between a compound c and a reaction r if c is present in the reaction

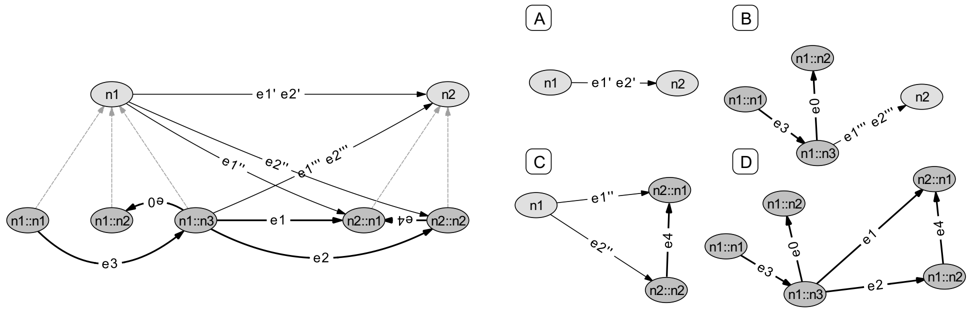


Figure 3.3: A simple hierarchical graph with two layers. L.h.s.: Dashed lines reflect the parent-child-relationship. Inserting the edge *e1* requires the construction of three virtual edges *e1', e1", e1"'*. Edge *e2* is already represented by two previously added virtual edges. Edges *e0*, *e3* and *e4* connect vertices having the same parent. No virtual edge is necessary. R.h.s.: Four expansion states of the graph (*n1* and *n2* collapsed [A], *n1* expanded and *n2* collapsed [B], *n1* collapsed and *n2* expanded [C], *n1* and *n2* expanded [D]).

r.

The layout algorithm allows multiple edges but no loops and proceeds recursively, starting with the top level graph and laying out a graph after its parent in the hierarchy has been laid out. For each graph, the layout consists of three phases:

Vertex Position places the vertices at crossing sections of a regular grid minimizing the stress,

Edge Routing places edges on a sequence of grid lines, minimizing a global edge cost function, and

Edge Bundling displaces edges by a small amount to avoid overlapping edge routes.

3.2.1 Vertex Position

In the VERTEX POSITION phase, we try to find a unique integer position for each vertex in a given *layout graph* (V, E) that minimizes the stress and therefore producing nicer images. The stress describes the amount of error that takes place by the projection of the *high-dimensional* graph-theoretic distances $d_{i,j}$ to the distances between the vertices' positions $\vec{x}_i, \vec{x}_j \in R^d$ (here $d = 2$):

$$stress = \sum_{i<j} d_{i,j}^{-\alpha}(d_{i,j} - ||\vec{x}_i - \vec{x}_j||_p)^2 \tag{3.1}$$

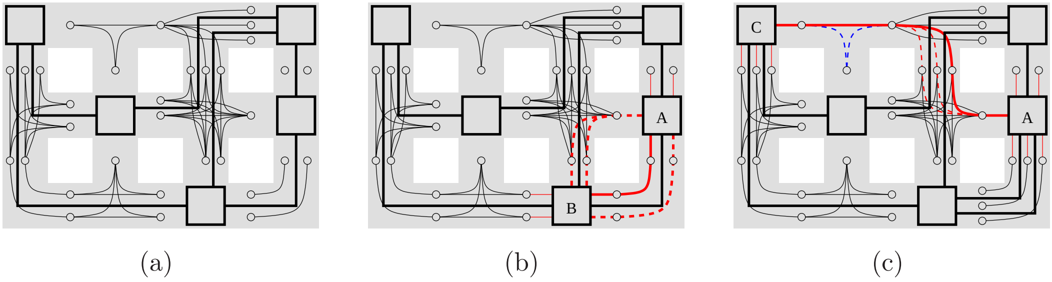
α controls the impact of (graph theoretic) distant vertices on the cost function. Larger values for α suppress the influence of distant vertices on the stress function. We choose $\alpha = 0$ to also consider vertices being more distant. p denotes the norm to use, $p = 2$ is the Euclidean distance, but as we are laying out the vertices and edges on a grid, we found it more natural to use the Manhattan distance, $p = 1$. If the graph consists of multiple connected components, we assume the graph theoretic distance between their elements to be $\sqrt{|V|}$ rather than infinity.

We position the vertices in a regular quadratic *grid graph* $G = (P, S)$, with P being at least $4 \cdot c \cdot |V|$ *positions*. c is the ratio of the number of available grid positions with respect to the number of graph vertices. We used $c = 4$ to give enough room for chain-like substructures to unfold. Throughout this phase, we restrict vertices to lie on even integer positions in both dimensions, the other positions are reserved for the edges' routes. The edges of the grid graph S are called *segments* and connect neighboring positions, i.e., positions that differ by an amount of one in exactly one coordinate. The *positioning* can be described as a function $p : V \rightarrow P$, and is *feasible* if no two vertices map to the same position.

We developed three algorithms for this phase. Each of them starts with a random, feasible positioning and iteratively improves the stress by finding a better feasible positioning.

A first *brute force* version selects a random vertex and puts it at the grid position that minimizes the local stress by testing each grid position. This has the nice property that the stress can never increase, because the vertex may always be put back at its original position. If the desired position is already taken by another vertex, their position is swapped.

As graph drawing by stress minimization was pioneered by *Kamada and Kawai* [36], we implemented a version of their algorithm using integer positions: we select a vertex with high local stress and, using a Newton-Raphson iteration, find a continuous position for that vertex where its local stress becomes minimal. We then find the closest grid position and insert it at that place.



(a) (b) (c)

Figure 3.4: A *layout graph* and its *route graph* during the insertion of two edges. The layout graph consists of the big square vertices and the thick bending lines. The grid is indicated by the shaded bars in the background. (a) The route nodes are drawn as small circles and the possible connections between them as thin curves. (b) To route an edge from A to B, edges to their neighboring route nodes are inserted first. Then, a SPSP algorithm finds the thick red path that is used for the edge route. The dashed red lines indicate alternatives. (c) When routing an edge from A to C, again there are several possible paths each one resulting in different crossing points of the new edge. The dashed blue deviation from the red path is illegal: the algorithm avoids it by prohibiting connections that belong to the same grid position to appear in direct succession on an edge route.

The third method selects a random vertex, inserts it randomly at a new position and then reevaluates the stress. Improvements are always kept. Should the stress deteriorate by Δ, it is only kept with a probability of $exp(-\Delta \cdot 1000/i)$ with i being the current iteration number. On rejection, the vertex is simply put back at its original position. This principle is known as *simulated annealing* [45].

All discussed algorithms terminate after a fixed number of iterations that is proportional to the number of vertices.

In the hierarchical version of the algorithm, edges connecting vertices of different pathway graphs will be specially treated. As the top level has already been laid out, it is known whether the edge enters a pathway vertex from the north, east, south, or west. For each of these directions, we add a *port vertex* centered on the boundary of the grid. A port channels all inter-pathway edges when entering a pathway from one of the four directions. The port vertex is not moved during the VERTEX POSITION phase, but is included in the stress minimization computation. This favors vertices incident on inter-pathway edges being closer to the port vertex.

3.2.2 Edge Routing

The EDGE ROUTING phase computes a combinatorial description of an edge routing along the segments (edges) of the *grid graph* leaving the vertices at their fixed positions computed by the VERTEX POSITION phase. It starts with the trivial edge routing for the layout graph containing only the vertices and successively inserts edge routes.

The routing of one edge given the fixed routing of all previous edges is computed by solving a Single Pair Shortest Path (SPSP) problem on a *route graph* $R = (N, C)$ and a function r, which yields for each grid segment a (possibly empty) ordered list of edges. For each grid segment s, R contains $|r(s)| + 1$ *route nodes* each one denoting a position between the existing routes on s. If no edges have been laid out yet, then this graph simply contains one vertex for each grid segment. A *connection* (u, v) exists in C if a route node u of a particular grid segment may be followed by the route node v of an adjacent grid segment in an edge route and vice versa. Conceptually, we could add all possible connections between route nodes that lie on adjacent grid segments, but we restrict the set of edges to avoid several undesirable configurations. Firstly, we do not want edge routes to lie on vertices they do not connect, and therefore, we only consider grid segments that are not incident on a position taken by a vertex. Furthermore, there may be no connection between route nodes of adjacent grid segments if there is an existing route using both segments and the route nodes lie on different sides. This ensures that edge routes can only cross at the points they meet or leave each other. Figure 3.4 shows the route graph for a small layout graph where some edges have already been laid out. Note that in that image, the vertices are not restricted to lie on positions divisible by 2 for illustrative purposes.

As there can be several possible routes for an edge, we have the ability to choose the one that gives the best visual impression. We found that we can simply assign weights to each route node and connection and perform a weighted SPSP algorithm to get the best route. The weight of a route path consists of four parts: its length, the sum of its segment densities, the number of edge crossings, and the number of bends. The length is simply the number of route nodes on the path. The density of each segment is the number of edges already routed using this segment. The edge crossing and number of bends can be computed for each connection before the SPSP is executed. Note that these costs never change during the SPSP as this would mean that an edge crosses itself or uses a segment more than once. Visually this would result in a cycle – an impossible output configuration of the SPSP algorithm.

To reduce runtime and memory consumption, we use the A^* search algorithm [30] to solve the SPSP instance. As a heuristic, we only use the Manhattan distance from the current route node to the destination and add 2 if we require at least one more bend. The crossing and population cost cannot be trivially approximated.

After all edges have been laid out, we try to further optimize the layout by removing edges of high cost and reinserting them again using the SPSP routing. Note that because of the optimality of the edge routing, the global cost can never increase. Each edge always has the possibility to use exactly the same route it had before.

3.2.3 Edge Bundling

The EDGE BUNDLING phase removes overlaps by shifting segments of edges' routes orthogonal to the grid segments they lie on. 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 (DAG) $D = (L, O)$ called the *displacement graph*. A *maximal line* of an edge route r is a maximal sequence of consecutive grid segments that lie on a line and that r uses. For each maximal line of an edge route, we add a vertex to L, and O contains an arc (l_1, l_2) if and only if the maximal lines l_1, l_2 share at least one grid segment and l_1 precedes l_2 in the relative ordering of routes on any shared segment. Because we disallowed parallel maximal lines to cross, the resulting displacement graph is acyclic. Figure 3.5 shows an example for the displacement graph of a set of maximal lines.

Any topological numbering of the displacement graph gives a displacement that avoids occlusions between edge routes of the same column/row. However, we use the topological numbering of minimum weight to get a displacement that packs lines closer together. Let $\delta : O \rightarrow R^+$ be the minimum separation between lines and $\omega : O \rightarrow R^+$ be the importance of an edge (for our purposes simply 2). The *topological numbering of minimum weight* $\lambda : L \rightarrow R$ can be solved by the

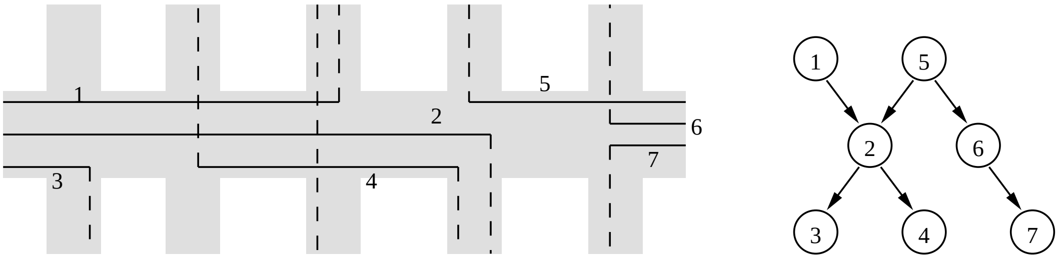


Figure 3.5: The displacement graph for a set of maximal lines. Actually, the transitive reduct is shown only.

following linear program:

$$
\begin{aligned}
min \sum_{(u,v)\in O} \omega(u,v)(\lambda(v) - \lambda(u)) \\
\text{subject to: } \delta(u,v) \leq \lambda(v) - \lambda(u) \quad \forall(u,v) \in O.
\end{aligned}
\tag{3.2}
$$

We note that the transitive reduct of the displacement graph suffices for our purposes. We use the same minimum separation for all pairs of lines and get the final displacements centering the lines by subtracting the output of the topological numbering by their barycenter weighted by line length.

3.2.4 Alternative Grid Generation

In the presented grid layout algorithm, the vertex placement and edge routing are performed in two separate consecutive steps. This offers the opportunity to develop alternative node placement routines fitting the specific needs of metabolic pathway visualization. Topological substructures, such as cycles or chains, are fairly common patterns in chemical reaction networks and should be taken into account when laying out pathway graphs. An algorithm for accomplishing this task is suggested in [6]. Metabolic network graphs provided by KEGG have already been laid out manually keeping these criteria in mind. As the drawings provided by KEGG constitute a de-facto standard in metabolic network visualization and biologists are familiar with that kind of representation, it is preferable to incorporate the layout information in our algorithm. Using the pathway data from KEGG has a particular advantage over other sources of metabolic data since the layout

is partially accessible. For the bottom-level graphs – reaction networks – node positions are given, however edge routes are not available.
As explained in the previous section, the vertex positioning phase of the algorithm generates a grid placing the vertices to discrete positions with no two vertices occupying the same location. The edge routing then takes this grid and vertex assignments as input. Instead of the aforementioned routine to assign vertices to grid cells, we iteratively generate a compact grid from given vertex positions in the following way:

1. Generate a sparse grid with as many columns as unique x values and as many rows as unique y values among the vertex positions in the 2D plane.
2. For each row i, we store a set of vertices U_i and for each column j a set of vertices V_j occupying that particular row or column. If no two nodes occupy the same point in the original 2D space, all grid cells will contain at most one node, i.e., $\forall_{i,j} |U_i \cap V_j| \leq 1$.
3. Merge $U_{i,i+1} := U_i \cup U_{i+1}$ with the shortest distance $|y_{i+1} - y_i|$ that causes no collision, i.e., for every column j we check the condition $|U_{i,i+1} \cap V_j| \leq 1$.
4. Merge $V_{j,j+1} := V_i \cup V_{j+1}$ with the shortest distance $|x_{j+1} - x_j|$ that causes no collision, i.e. for every row i we check the condition $|V_{j,j+1} \cap U_i| \leq 1$.
5. Go back to step 3. Repeat until no rows or columns can be merged without violating the collision condition.

When merging the vertex sets of two adjacent columns, the new x coordinate of all vertices in that union is the arithmetic mean value of the x coordinates of the two adjacent columns. The same holds for the y coordinate of vertices in the set union after merging two adjacent rows. In a sense we perform a one-dimensional k-means clustering on the x and y coordinates separately to achieve a compact grid. This algorithm ensures that the directional relations of the node positions are preserved. In other words, if *node A* was "above" *node B* before the grid computation, *A* will never be "below" *B* on the resulting grid. The discretization of the node positions preserves the mental map as much as possible. Figure 3.6 depicts the iterative grid generation. Setting appropriate constraints for the following edge route computation, the final drawing closely resembles the pathway presentations provided by KEGG (see Figure 3.7). We found that penalizing edge bends and favoring edge bundles in the edge routing phase yields to the most appealing results.

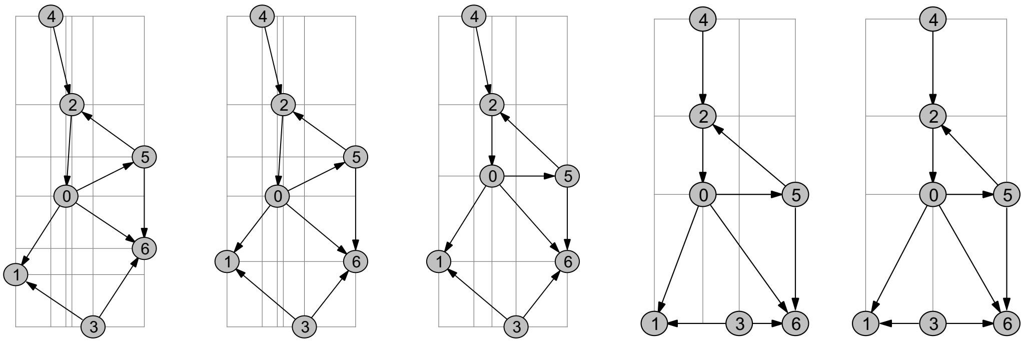


Figure 3.6: Iterative grid generation using original node positions. L.h.s.: sparse grid before the first merge step. The algorithm starts with a 6-by-7 grid for seven nodes. In the first step nodes 1 and 6 are merged into one set containing the nodes of row 5 and 6. The last step merges columns 2 and 3 resulting in a compact 3-by-4 grid. The algorithm terminates because any further merge operation violates the collision condition.

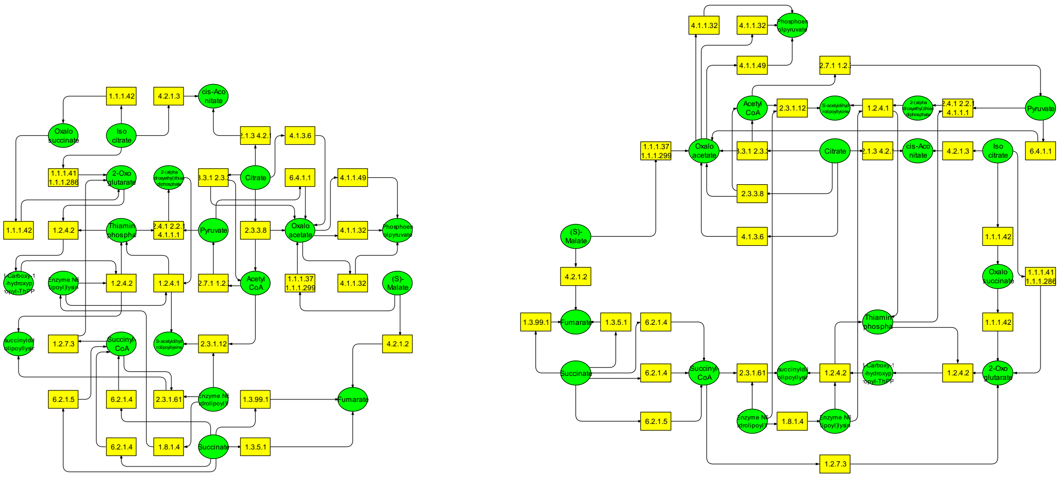


Figure 3.7: TCA cycle. L.h.s.: de-novo layout computation using our grid layout algorithm for node placement. R.h.s.: Grid generated from node positions of the KEGG layout.

3.3 Graph Exploration

The graph interaction and exploration methods described in this chapter 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 first present this technique with supplementary search and highlighting operations and explain later how the graphical user interface of the program lets the user apply these methods to interact with the graphical representation for exploring the metabolic network graph.

3.3.1 Hierarchical Graph Navigation

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 [60] applied to hierarchical graph exploration fulfills this requirement. It is an established focus&context method to give an overview on large tabular datasets to visually examine data 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 Section 3.2.1) 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 [69]. 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 or by creating an iconized view of the node's nested graph (see Figure 3.8).

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 layout.

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 Figure 3.9 and 3.10).

3.3.2 Semantic Zoom

In contrast to the conventional geometric zoom, which simply scales the graph drawing and therefore changes the size of the viewport, the semantic zoom increases the amount of detail being displayed while zooming into the scene. It can be considered as a form of 'details on demand'. Detailed information not relevant to the global view on the data is hidden at large scales, but can be made visible as the scene is enlarged and additional drawing space is provided. In current applications for the visualization of general graphs, this technique is often limited to hiding

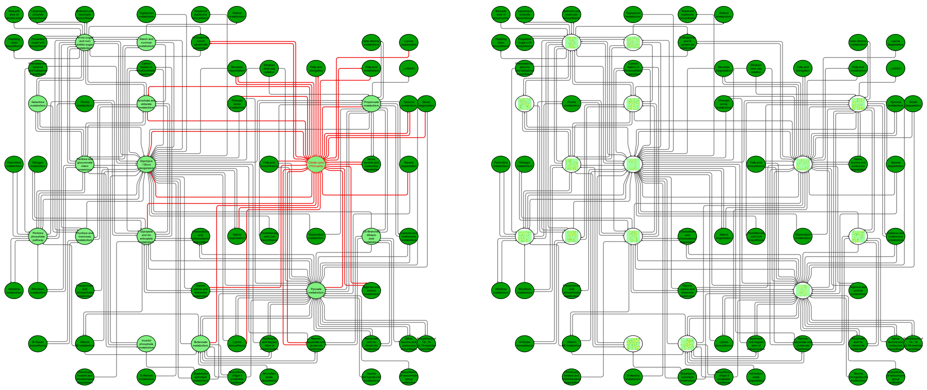


Figure 3.8: The top-level graph of the *Carbohydrate Metabolism*. Pathway nodes drawn in bright green color belong to the carbohydrate metabolism and can be expanded to reveal the reaction networks. Non-expandable related pathways are shown in dark green. L.h.s.: The pathway *Citrate Cycle* was selected and all incoming and outgoing connections are highlighted. R.h.s.: Icons within the pathway nodes represent the nested graph.

or displaying textual information in node or edge labels. In the context of metabolic network visualization, we included a renderer for molecule structures drawn inside nodes representing a chemical compound. As the user increases the level of detail, the chemical structures of metabolites is rendered within the associated nodes. We offer five levels of detail for rendering compound nodes:

1. Totals formula specifying the numbers for each atom present in the molecule (see Figure 3.11 l.h.s.).
2. Totals formula and systematic name of the compound.
3. Backbone or skeletal graph. Only the bonds between atoms are shown (Figure 3.11 center).
4. Molecular graph depicting the Fischer projection [53] of the molecule .
5. Molecular graph with all atoms (including inner Carbon and Hydrogen atoms, see Figure 3.11 r.h.s.).

The molecule structures can be considered as graphs, in which the vertices represent the atoms and the edges represent the covalent bonds between two atoms.

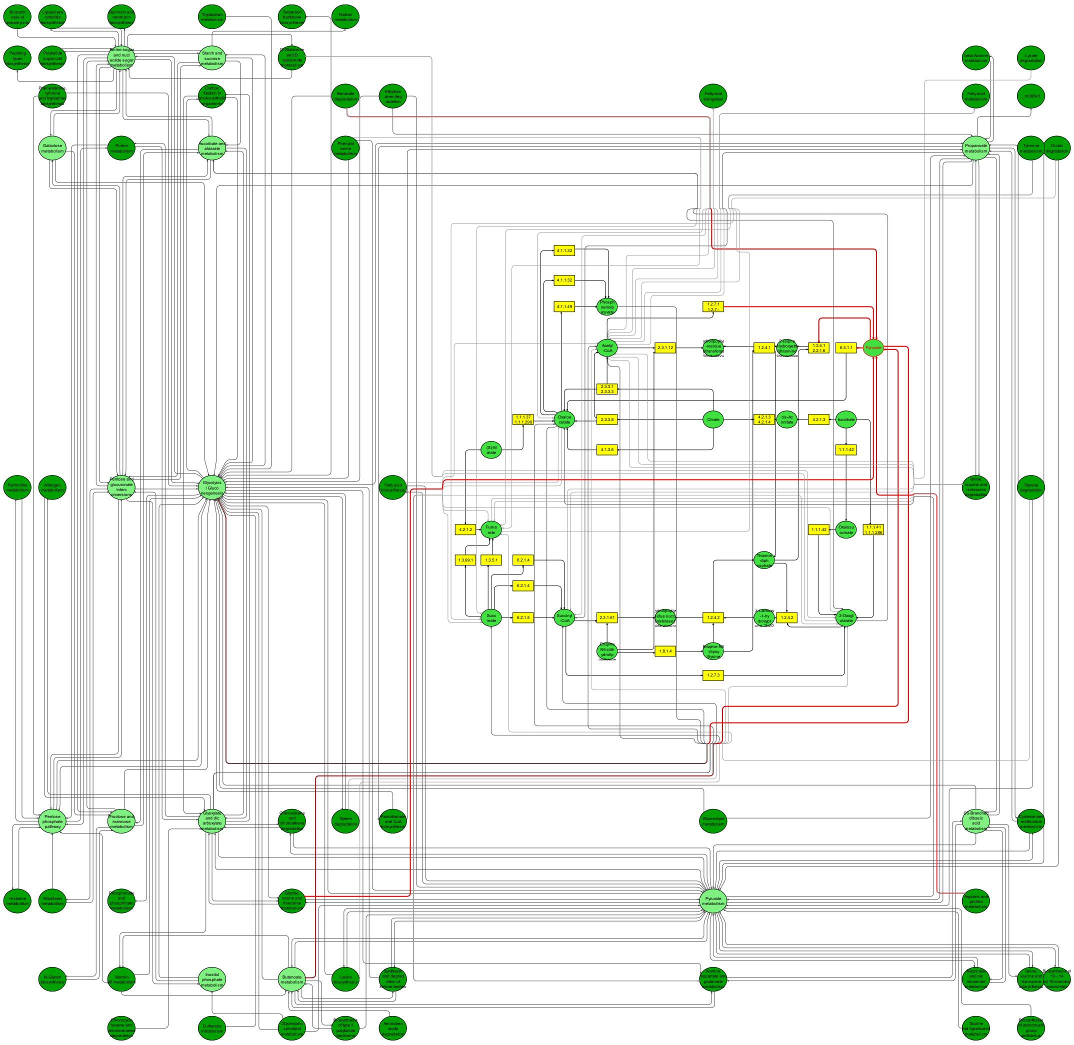


Figure 3.9: Detailed view of the expanded node *Citrate Cycle (TCA)*. The highlighted compound node *Pyruvate* establishes several connections to adjacent pathways.

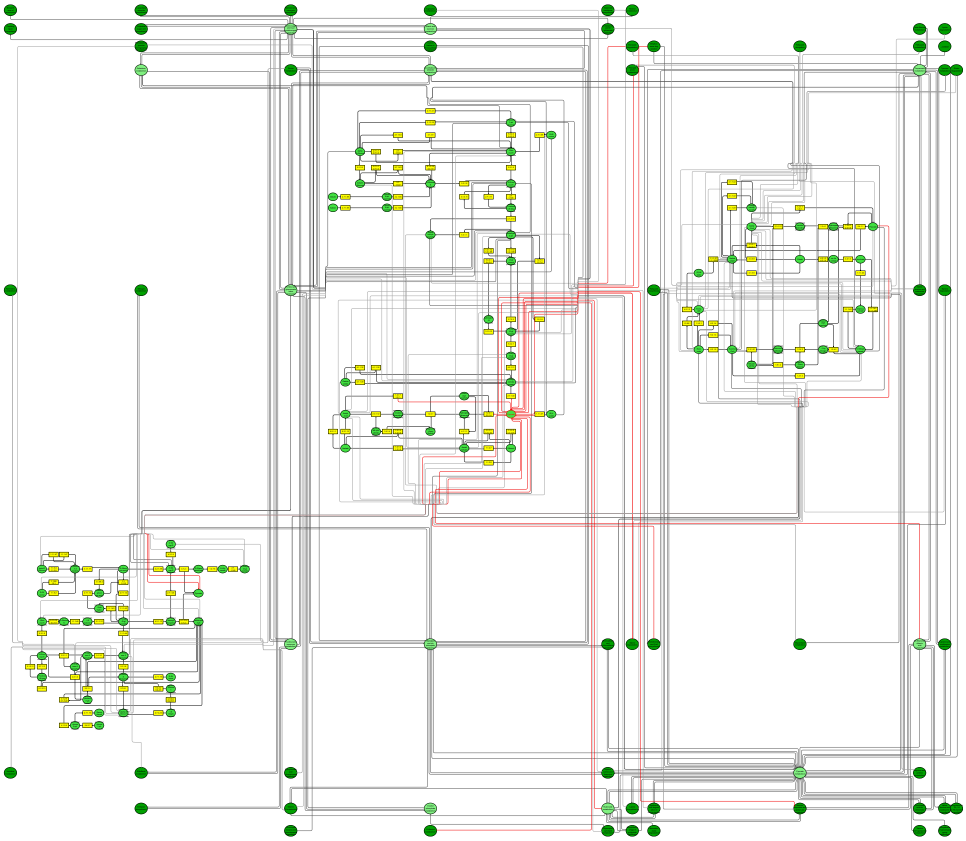


Figure 3.10: Multiple foci in the network visualization. Three expanded pathway nodes: *Citrate Cycle (TCA)*, *Pentose Phosphate Pathway*, and *Glycolysis/Gluconeogenesis*. The compound node *Pyruvate* within the Glycolysis/Gluconeogenesis pathway is highlighted showing the connections to adjacent pathway nodes.

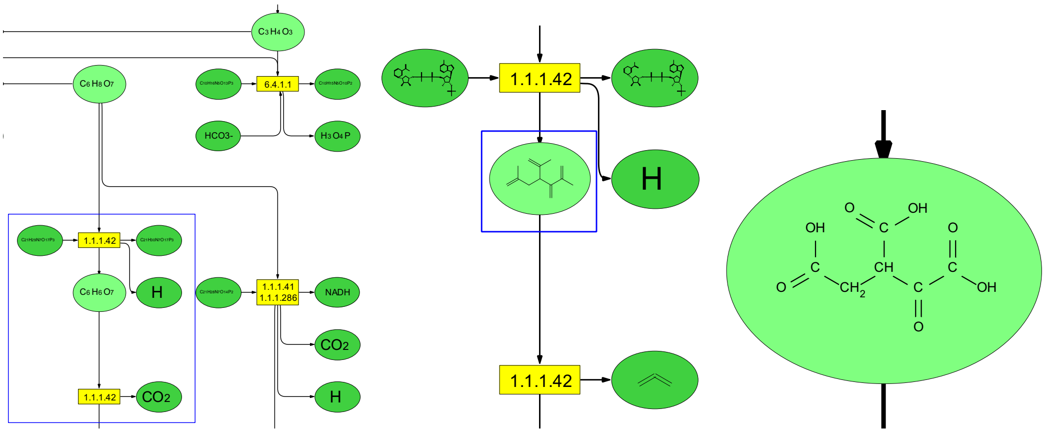


Figure 3.11: Semantic Zoom: Three views on some reactions of the Citric Acid Cycle (TCA). While zooming into the drawing of the metabolic pathway, the totals formula (l.h.s.), the backbone of the chemical compound (center), and the full molecule including all atoms present in the molecule are shown inside the compound nodes. The blue rectangles indicate the size and position of the viewport of the following image.

The molecule graph and its drawing are provided by KEGG via *MDL mol-files*. *MDL mol* is a common format for the representation of biochemical compounds. A description of the format can be found in [2].

3.3.3 Selection and Highlighting

The exploration of a large metabolic network is a highly interactive process as suggested in the previous sections. The hierarchical navigation of the network is a straightforward process and intuitive in its application, however the method has an inherent drawback. As the user expands pathway nodes, the total space needed for drawing the network can become very large. The worst case scenario arises when expanding the diagonal elements of the grid. In terms of the Table Lens metaphor, expanding a cell requires the column and the row in which the cell is located to be enlarged making room for the detailed network. External connections to compound nodes in the focused pathway have to be alongated making it difficult to discern their origin in adjacent pathways. To alleviate this problem, we provide several methods to highlight nodes and incident edges:

- Selecting a top-level node in the *Data Browser* (see Figure 3.13) highlights all nested elements including incident inter-pathway edges.

- Selecting a bottom-level node (chemical compound or reaction) in a particular pathway highlights the corresponding item in the *Graph Scene* together with its incident edges.
- A search mask allows the user to select items by performing a string-based pattern matching among the textual attributes of the graph elements highlighting the results in the scene.
- The selection of bottom-level nodes is propagated to the top-level of the graph to display search results regardless of the network's expansion state.

The search function is an intuitive way to state queries as "*Select all pathways containing the compound* Pyruvate" (see Figure 3.12).

The implementation of the Graphical User Interface follows the linked view paradigm. In this respect, the *Data Browser* and the *Graph Scene* components represent different views on the underlying graph model of the network. Selecting items in either one of the views instantly updates the selection state in the other. A more in-depth description of the developed software can be found in chapter 6.

3.4 Performance Results

The KGML import routine is suitable to construct pathway graphs of different size and complexity. To implement, test and demonstrate the discussed visualization and exploration techniques, we constructed two pathway graphs. Images 3.8 through 3.12 were created from 15 KGML 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 by *maplink* elements. A graph was created with a total of 640 compound nodes, 67 pathway nodes (52 non-expandable), 894 reaction hyperedges and 466 regular inter-pathway edges. Even though we model the pathway graph as a directed hypergraph, the proposed layout algorithm deals with regular graphs. 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)

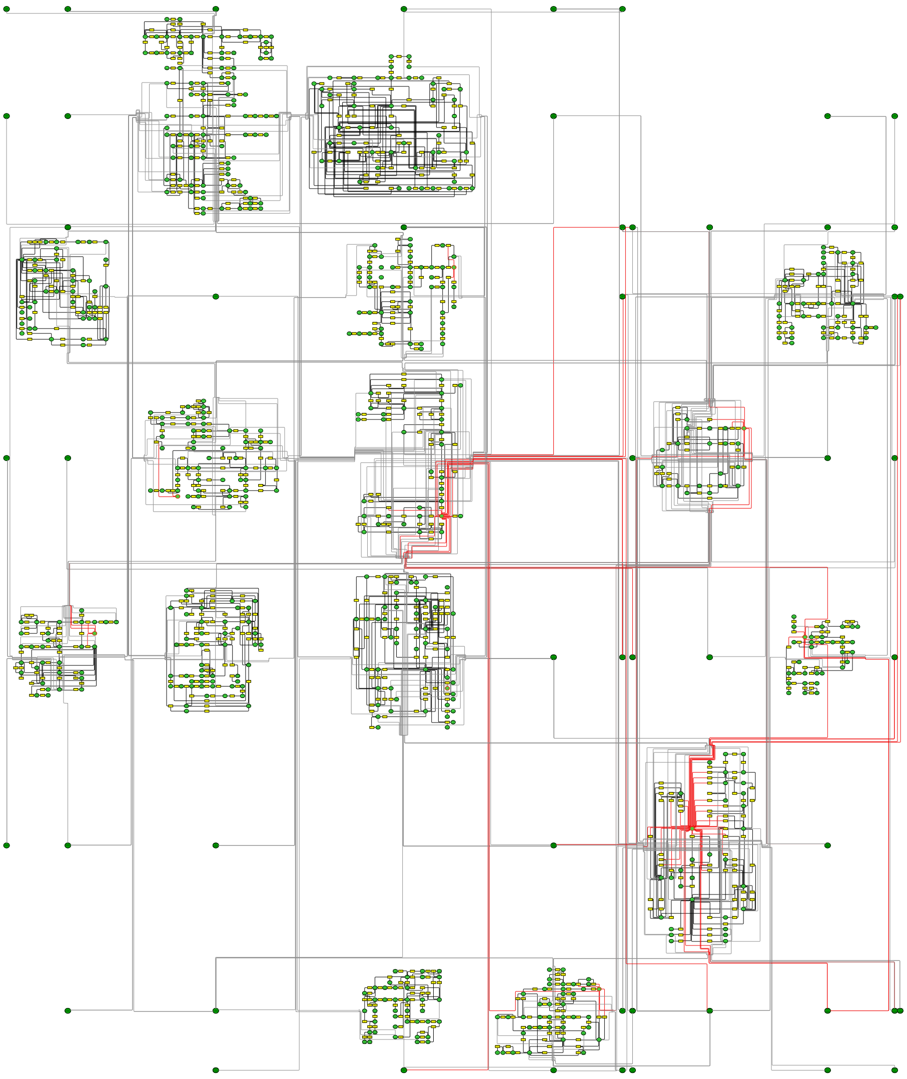
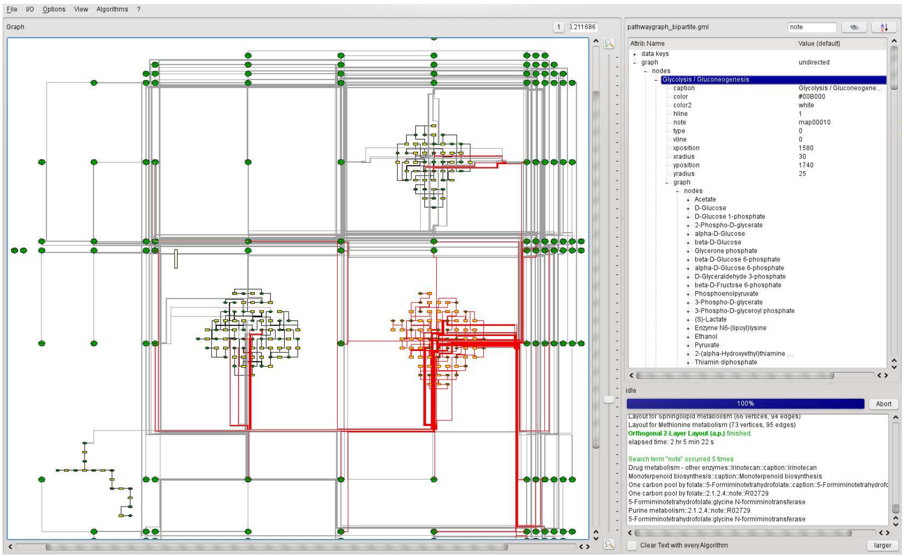


Figure 3.12: The bottom-level graph. Reaction network of pathways associated with the *carbohydrate metabolism*. The search result for *Pyruvate* is highlighted including its incident edges.

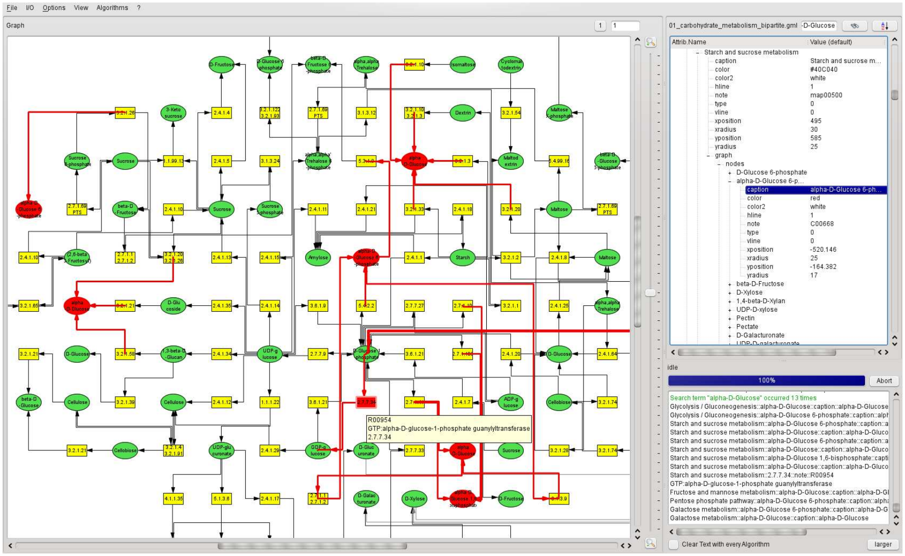


```
pathwaygraph_bipartite.gml                note
Attrib.Name                               Value (default)
+ data keys
- graph                                   undirected
  - nodes
    - Glycolysis / Gluconeogenesis
        caption                           Glycolysis / Gluconeogene...
        color                             #00B000
        color2                            white
        hline                             1
        note                              map00010
        type                              0
        vline                             0
        xposition                         1580
        xradius                           30
        yposition                         1740
        yradius                           25
      - graph
        - nodes
          + Acetate
          + D-Glucose
          + D-Glucose 1-phosphate
          + 2-Phospho-D-glycerate
          + alpha-D-Glucose
          + beta-D-Glucose
          + Glycerone phosphate
          + beta-D-Glucose 6-phosphate
          + alpha-D-Glucose 6-phosphate
          + D-Glyceraldehyde 3-phosphate
          + beta-D-Fructose 6-phosphate
          + Phosphoenolpyruvate
          + 3-Phospho-D-glycerate
          + 3-Phospho-D-glyceroyl phosphate
          + (S)-Lactate
          + Enzyme N6-(lipoyl)lysine
          + Ethanol
          + Pyruvate
          + 2-(alpha-Hydroxyethyl)thiamine ...
          + Thiamin diphosphate
idle
100%
```

```
Layout for Sphingolipid metabolism (66 vertices, 94 edges)
Layout for Methionine metabolism (73 vertices, 95 edges)
Orthogonal 2-Layer Layout (a.p.) finished.
elapsed time: 2 hr 5 min 22 s

Search term "note" occurred 5 times
Drug metabolism - other enzymes::Irinotecan::caption::Irinotecan
Monoterpenoid biosynthesis::caption::Monoterpenoid biosynthesis
One carbon pool by folate::5-Formiminotetrahydrofolate::caption::5-Formiminotetrahydrofc
One carbon pool by folate::2.1.2.4::note::R02729
5-Formiminotetrahydrofolate:glycine N-formiminotransferase
Purine metabolism::2.1.2.4::note::R02729
5-Formiminotetrahydrofolate:glycine N-formiminotransferase
```

Figure 3.13: Graphical User Interface 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.



```
R00954
GTP:alpha-D-glucose-1-phosphate guanylyltransferase
2.7.7.34
```

```
01_carbohydrate_metabolism_bipartite.gml  -D-Glucose
Attrib.Name                               Value (default)
  - Starch and sucrose metabolism
      caption                             Starch and sucrose m...
      color                               #40C040
      color2                              white
      hline                               1
      note                                map00500
      type                                0
      vline                               0
      xposition                           495
      xradius                             30
      yposition                           585
      yradius                             25
    - graph
      - nodes
        + D-Glucose 6-phosphate
        - alpha-D-Glucose 6-p...
            caption                       alpha-D-Glucose 6-ph...
            color                         red
            color2                        white
            hline                         1
            note                          C00668
            type                          0
            vline                         0
            xposition                     -520.146
            xradius                       25
            yposition                     -164.382
            yradius                       17
        + beta-D-Fructose
        + D-Xylose
        + 1,4-beta-D-Xylan
        + UDP-D-xylose
        + Pectin
        + Pectate
        + D-Galacturonate
        + UDP-D-galacturonate
idle
100%
```

```
Search term "alpha-D-Glucose" occurred 13 times
Glycolysis / Gluconeogenesis::alpha-D-Glucose::caption::alpha-D-Glucose
Glycolysis / Gluconeogenesis::alpha-D-Glucose 6-phosphate::caption::alph
Starch and sucrose metabolism::alpha-D-Glucose 6-phosphate::caption::a
Starch and sucrose metabolism::alpha-D-Glucose::caption::alpha-D-Gluco
Starch and sucrose metabolism::alpha-D-Glucose 6-phosphate::caption::a
Starch and sucrose metabolism::alpha-D-Glucose::caption::alpha-D-Gluco
Starch and sucrose metabolism::alpha-D-Glucose 1,6-bisphosphate::capti
Starch and sucrose metabolism::alpha-D-Glucose::caption::alpha-D-Gluco
Starch and sucrose metabolism::2.7.7.34::note::R00954
GTP:alpha-D-glucose-1-phosphate guanylyltransferase
Fructose and mannose metabolism::alpha-D-Glucose::caption::alpha-D-Gl
Pentose phosphate pathway::alpha-D-Glucose 6-phosphate::caption::alpha
Galactose metabolism::alpha-D-Glucose 6-phosphate::caption::alpha-D-G
Galactose metabolism::alpha-D-Glucose::caption::alpha-D-Glucose
```

Figure 3.14: 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/or displays results from a keyword search within the data browser. In this scenario a search for the term *alpha-D-Glucose* was performed resulting in 13 matches within the pathways related to Carbohydrate metabolism. The Data Browser also highlights the matched items within the hierarchical representation of the dataset.

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 1,601 nodes and 2,505 edges. This graph could easily be handled by the visualization software. 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. The graph covers the complete metabolism of a generalized eucaryotic cell and contains 4,980 compound and 145 pathway nodes, 4,943 reactions and 1,248 inter-pathway edges. After the conversion to the corresponding bipartite graph, the network 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 program was tested on a machine with a Intel(R) Xeon(R) CPU at 2 GHz and 32 GB RAM.

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 and chain-like structures. This is due to the larger space available to unfold those substructures. However, a higher number of potential node positions increases the runtime for the brute force method.

3.5 Conclusion

With the development of our graph visualization and exploration framework, we are able to layout and display complex graphs with a high number of edges and nodes. For metabolic pathway networks in particular, not only the graph topology is relevant, a high number of additional attributes – mainly textual annotations, but also hierarchical and structural information – has to be considered in the visualization, too. This leads to different scales in granularity. We can conclude that the exploration of static metabolic networks containing different levels of detail, specifically the pathway level, reaction level, and molecular level, is a highly interactive process.

Semantic zooming and focus&context methods are applied to accomplish this task, 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 currently available lack this key feature. Even though node expansion and collapse produce very discrete and rather abrupt changes in the appearance of the graph, only the row and the column of the grid position are affected. All remaining elements keep their relative position to each other. 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. A simple extension may be to perform node expansion and collapse in a semi-continuous manner by interpolating node and edge route positions between the two states thus creating the effect of smooth and gradual changes to the graph layout. This will support maintaining the mental map and simply look appealing. Of course, this should be verified by user studies and/or evaluations.

4 Visualization of Dynamic Graphs

We extend our previous work on the exploration of static metabolic networks to evolving, and therefore dynamic, pathways. We apply our visualization framework to data from a simulation of early metabolism. Thereby, we show that our technique allows us to test and argue for or against different scenarios for the evolution of metabolic pathways. Virtually, no aspect of the system under research is left unconsidered due to the versatile points of view and high scalability that is provided. This supports a profound and efficient analysis of the structure and properties of the generated metabolic networks and its underlying components, while giving the user a vivid impression of the dynamics of the system. The analysis process is again inspired by Ben Shneiderman's mantra of information visualization. For the overview, user-defined diagrams give insight into topological changes of the graph as well as changes in the attribute set associated with the participating enzymes, substances and reactions. This way, "interesting features" in time as well as in space can be recognized. A linked view implementation enables the navigation into more detailed layers of perspective for in-depth analysis of individual network configurations.

4.1 Evolution of Metabolic Networks

Metabolic networks, the set of chemical compounds and their interactions that constitute life in the most basic sense, are the best studied biological networks. With a wide availability of genomic, proteomic and metabolomic data it becomes possible to study cell behavior. However, to understand the underlying principles of life and gaining further insights about the metabolism of cells for the use in biotechnological applications, e.g., pharmaceutical target prediction or metabolic engineering, we need tools to model and analyze the metabolic processes, pathways, and networks. There exist successful means for the reconstruction of metabolic networks from annotated genomes [58], the analysis of these networks in terms of elementary pathways [24], and description of their behavior with the help of ODE models [78]. Further insight into the development of kinetic models of metabolic networks addressing rate laws of the involved enzymes is provided in [70]. The situation becomes more difficult when we want to explain the evolutionary mechanisms of these systems, i.e., the formation of metabolic pathways or the emergence of complex network properties. Although, several scenarios exist that provide some insight into the evolution of metabolic pathways [11], only few aspects are well understood. Especially, the first steps in early metabolism evade observation by conventional approaches. To this end, Ullrich et al. [73] developed a multi-level computational model to study the transition to life: the evolution of metabolic pathways from catalyzed chemical reactions. The simulation approach implements components on different scales in a more realistic manner than has been done so far.

In this work we introduce a plug-in for exploring dynamic graphs extending the existing graph visualization software described in the previous chapter. The implementation of the extension was primarily driven by the given data and the requirements stated by the scientists providing it. These include

1. Overview of the complete series of evolving metabolic networks, i.e., involvement of metabolites, reactions and enzymes, and evaluation of key properties, e.g. quantity (concentration) and activity (participation in pathways).
2. Analysis of dynamics in the network's topology and attribute set. Compare networks of different time steps and analyze topology dynamics in more detail.
3. Elementary pathway analysis of selected network generations.
4. Time series analysis of attributes associated with selected node.

For the analysis of the simulation results, an efficient visualization system tailored to suit our needs is of utmost importance. The main function of the software introduced in this article lies in the analysis of metabolic networks in general and studying the evolution and dynamic behavior of metabolism in particular. This is achieved by providing an insightful overview on different scales (e.g., on the metabolite-, pathway-, or network-level) and different angles (e.g., dynamics in topology vs. attribute dynamics) of the vast amount of extracted information. Being able to observe all components (individually or together) for the entire simulation time in one representation gives us a much deeper understanding of the system's dynamics than any statistical analysis or static view can provide. By means of one sample simulation, we show the possibilities of the method and which potential general insights we can gain.

4.2 The Model

In this section we introduce a computational model of early metabolism for studying the emergence and evolution of catalyzed chemical reaction networks. The model consists of a graph-based artificial chemistry allowing for realistic kinetic behavior, and a protocell-like entity that inhabits the artificial chemistry and that is exposed to changes (e.g., mutations, source) and selection against other protocells.

The artificial chemistry of this model is motivated by the chemist's intuition of molecules and chemical reactions. Consequently, molecules are modeled as labeled graphs, with atoms as nodes and bonds as edges. Given this representation, it is easy to see that chemical reactions can be understood as graph transformations, or in computer science terms, as simple graph rewriting rules. Metabolic networks are expanded using a stochastic network generator inspired by Faulon [21]. For simplicity, reaction rates were computed here based on topological indices (Wiener number [77]) of the educt and product molecules of the reactions.

The protocell depicted in Figure 4.1 contains a simple cyclic genome with several RNA-genes encoding for a particular reaction type (graph rewriting rule) through a sophisticated genotype-phenotype mapping [74]. The genome is subject to mutation, deletion, duplication and horizontal gene transfer events. Therefore, reactions can occur, change and disappear from the protocell or even get copied to a neighbor. In each generation, only half of all protocells is selected and generates an identical copy. There is steady influx of metabolites from the environment and out

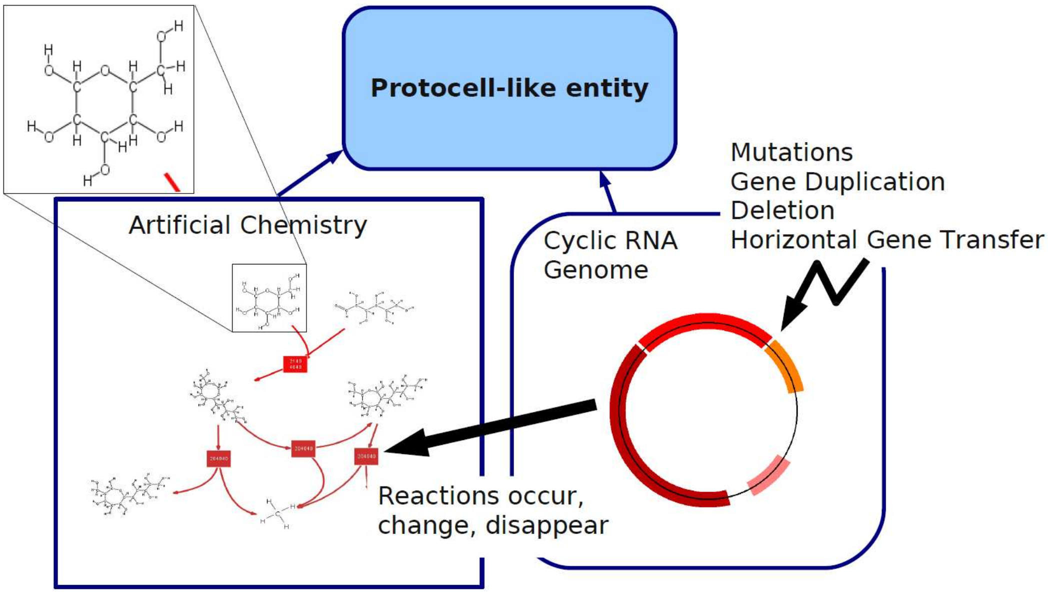


Figure 4.1: The protocell in the simulation consists of a cyclic RNA genome encoding rewriting rules applied to the graphs representing the cell's artificial chemistry.

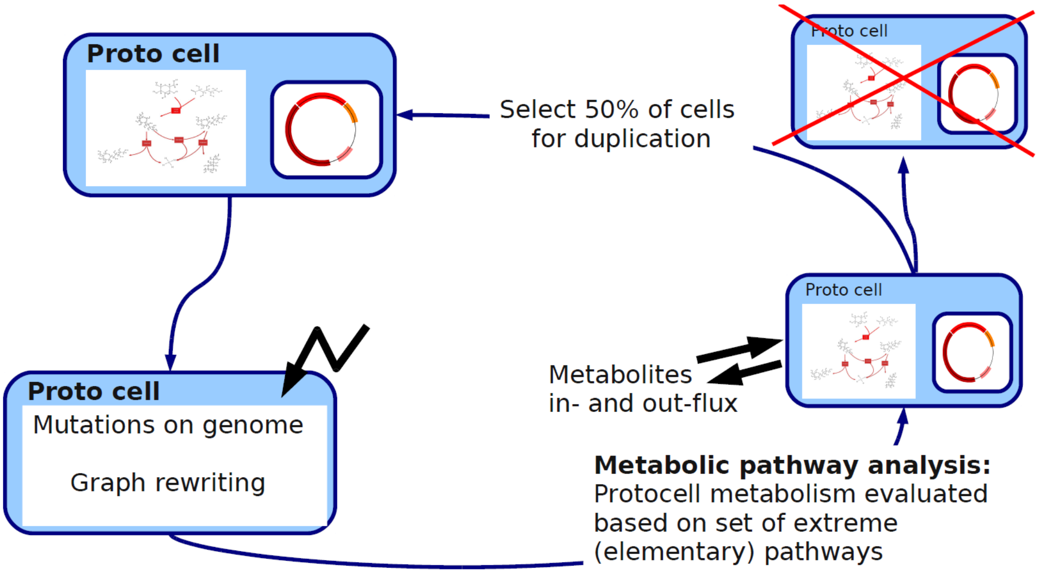


Figure 4.2: One iteration of the simulation run. After introducing mutations on the RNA genome, the genome-function mapping determines how the molecular structure graphs are rewritten. The metabolic pathway analysis decides the cell's survival for the next iteration.

flux of produced metabolites in way of biomass production. The constitution of either may change during the course of the simulation.

The metabolism of a protocell is evaluated based on its metabolic yield, which can be determined by the set of extreme (elementary) pathways [58] in a metabolic network. We use metabolic pathway analysis to compute this set of elementary pathways from the stoichiometric matrix by fulfilling the steady-state constraint as well as all inequality constraints.

4.3 The Data

In the analysis of the simulation results, several types of information on different levels are processed. Most importantly, the structure of the metabolic network in form of a bipartite labeled graph is stored in a GraphML [9] file. Metabolites and enzymes are the nodes of the graph, while reactions are represented by edges. The labels for enzymes and reactions are unique identifiers giving insight to their function. The metabolite label is its canonical SMILES strings [76], a unique structural representation that is easily readable for chemists. Further, the concentration (number of molecules) for each metabolite is included in the network information. In addition to the network information, flux information – the set of elementary pathways through the network – is made available to the visualization plugin, in a simple text file. Extremal nodes are listed. These represent the metabolites transferred into the cell and those that are used as biomass or excreted into the environment, respectively. For each reaction it is noted whether it is present in a particular elementary pathway or not (0 or 1).

All types of information are generated for each generation. Since the simulation has several parameters and input options, the data can be very diverse in size and number of files as well as complexity. Here lies also one important merit of the visualization method: Choosing an "interesting" simulation run for further analysis from the range of possible simulations. The visualization of all levels and generations combined allows an efficient decision process that is of particular importance in a development and testing stage.

In the generated data, each enzyme catalyzes an arbitrary number of chemical reactions. We therefore represent a metabolic network as a hierarchical bipartite

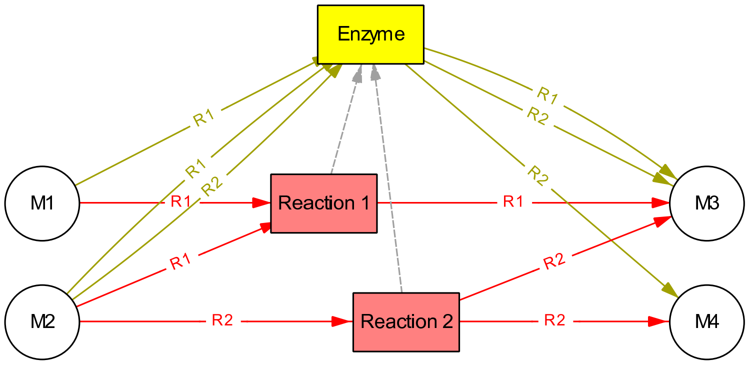


Figure 4.3: One enzyme node can have an arbitrary number of reaction nodes as children. Metabolites (circles) are un-grouped. Either the red elements (Reaction View) or the yellow elements (Enzyme View) are visible. Metabolites are always shown. The dashed lines indicate the child-parent-relationship.

graph, where reaction nodes have exactly one associated enzyme node as parent (see Figure 4.3).

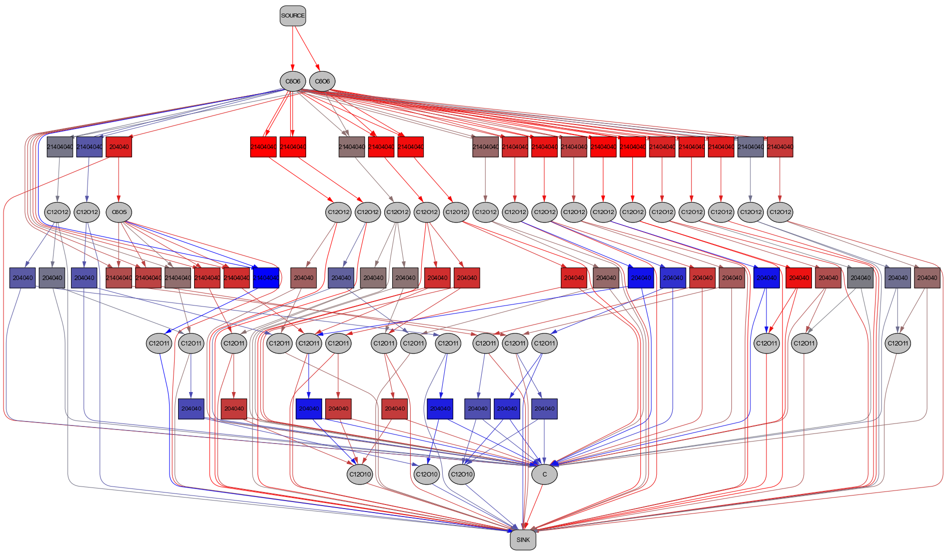
4.4 Visual Data Exploration

In this section we focus on different visualization techniques implemented to support the data analysis process. Based on the hierarchical nature of the graph as described in the previous section, we implemented two different views on the metabolic network graph:

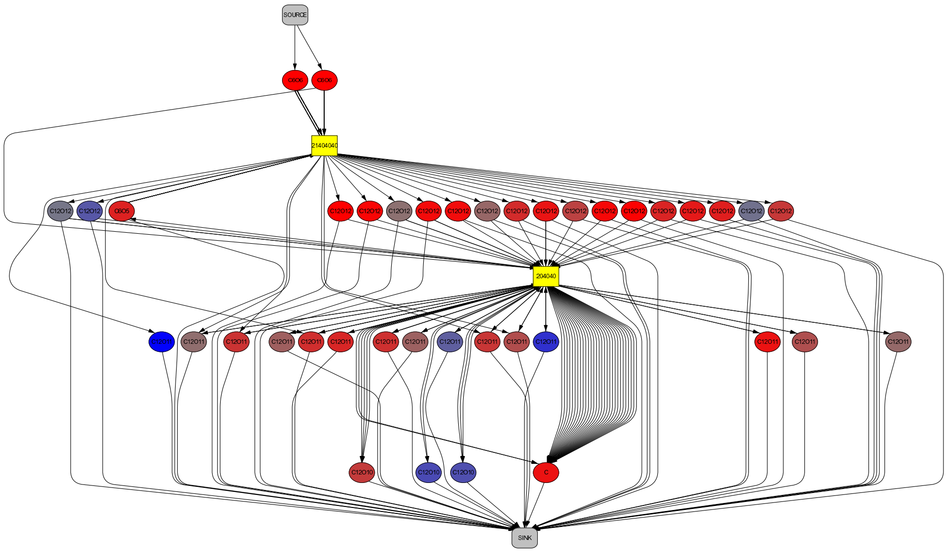
- *Reaction View*, Figure 4.4(a): Showing the bottom-level of the graph by expanding all enzyme nodes reveals the associated child nodes (reactions) and hides the enzymes with their incident edges.

- *Enzyme View*, Figure 4.4(b): Showing the top-level of the graph by collapsing all enzyme nodes hides the associated child nodes.

During the data exploration process, the user may always switch between the two views without changing the actual topology of the graph. In this context, mental map preservation is a key requirement for analyzing dynamic networks [51]. Furthermore, changes in the graph drawing from one network generation to the next should be minimal if the topological changes are small. "Jumping nodes" should therefore be avoided. We put special emphasis on that issue when switching



(a) Reaction View



(b) Enzyme View

Figure 4.4: Union Graph laid out using the Sugiyama layout algorithm. The reaction nodes (rectangles) are colored according to their first appearance (red: earlier, blue: later). Note that the positions of metabolite nodes (ellipses) remain the same. In the enzyme view, we apply the node coloring scheme to metabolites instead of reactions.

between reaction and enzyme view, and browsing the time line. Preserving the mental map is also crucial for producing animations of the network evolution.

We achieve the requirement of mental map preservation by following the idea of [17] and create a foresighted layout by constructing the *Set Union Graph* or *Super Graph* $\hat{G} = (\hat{V}, \hat{E})$ with $\hat{V} = \bigcup_{i=1}^{n} V_i$ and $\hat{E} = \bigcup_{i=1}^{n} E_i$ where $(V_i, E_i) = G_i$ is the network after i generations. After the preceding cycle removal, we lay out $\hat{G}$ using Sugiyama's method for directed acyclic graphs [72, 25]. This layout method is suitable for our visualization, because the constructed graph contains very few cycles, and the general direction of fluxes through the network is suggested by the graph drawing, i.e., from top (source) to bottom (sink). To emphasize the importance of extremal nodes – metabolites existing in the cell with no reaction producing them (source metabolites) and metabolites with no reaction consuming them (biomass production) – we connect them to a global source or sink node, i.e, the resulting acyclic graph becomes a so-called *st-graph* [15].

The super graph contains elements of the reaction view and the enzyme view at the same time. Layouting this graph ensures the metabolite nodes' positions to remain constant when switching between the views or changing to a different network generation.

The three requirements stated in the Introduction meet Ben Shneiderman's mantra of information visualization [69]. In the following we describe the visual analysis process based on the scheme *"Overview first, zoom and filter, details on demand"*.

4.4.1 Overview

After construction of the Set Union Graph and associating the flux information with the graph elements, the primary objective of the overview visualization is to give the user a general idea of the network elements – metabolites, reactions and enzymes – involved, their life time, and the development of fundamental attributes associated with the network elements over time (see Figure 4.5). When presenting the Set Union Graph (Figure 4.5a), a given node coloring scheme distinguishes between older and newer nodes. The time of first occurrence of a node in the network determines its color. The node appearing first is red, the node appearing last is blue. Node colors in between are interpolated using the color scale depicted in Figure 4.10c, third from left. The user may choose, whether this scheme is applied to reaction nodes, to molecule nodes, or both. Further insight into the life times of metabolites and reactions give the interval diagrams depicted in Figures 4.5(f, g).

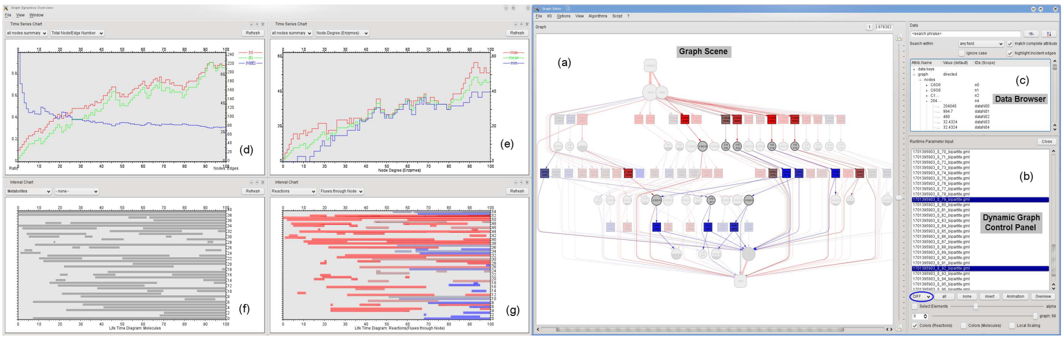


Figure 4.5: Graphical User Interface of the Dynamic Graph Analysis plug-in. *Overview visualization*: Time Series Charts of selected attributes (d, e) display attribute dynamics over time. Interval Charts (f, g) represent the dynamic topology of the graph in terms of life times of metabolites, enzymes, and reactions. In (g), horizontal bars depicting the nodes' life time have been overlaid with the attribute *Fluxes through node*. The Graph Scene (a) shows the Set Union Graph with the applied node coloring scheme. As for *Zoom and Filter*, the user may select different network generations in the *Dynamic Graph Control Panel* (b) to apply the set operators DIFF, AND, OR for filtering certain elements. These will be assigned a user-defined alpha value and/or highlighted for selection.

Except for the artificially inserted environment nodes (global source and sink), each row represents a node in the graph. Horizontal bars depict the life time and may be overlaid with additional quantitative information, e.g., node degree, fluxes through that node, and concentration for metabolite nodes, which is depicted as a curve inside the corresponding interval bar. In addition to interval diagrams, time series charts (4.5d, 4.5e) summarize selected attributes and display their dynamics over time. The user can again choose the subset of nodes to be taken into account (metabolites, enzymes, or reactions) and the attribute to be visualized (node number, node degree, number of elementary fluxes through a node, and concentration values), and combine these time series in any way for comparison.

4.4.2 Zoom and Filter

In this analysis step, the user wants to detect "interesting features" in the overview and select individual networks for further inspection. Interesting in an evolving metabolic network may be periods of stabilities or instabilities in a topological sense – appearance of new reactions or metabolites – as well as in terms of flux behavior – changes of associated attributes.

The straightforward approach is to simply browse the time line. For that purpose, we have implemented a linked view connecting the diagrams of the overview visualization with the dynamic graph in the Graph Scene. The screen shot of the software given in Figure 4.6 gives an impression on that type of navigation. The user may jump directly to that time point of interest by clicking into any of the displayed diagrams to further inspect the associated network. For each point in time, the current attributes are visualized in the nodes. For metabolites, the concentration values are depicted by the "filling level" of the node. Additionally, the node sizes and edge widths represent the number of elementary pathways these elements participate in. At this point, the user can control the scaling of edge widths and node sizes (see Figure 4.7). Controlling the scaling of network attributes with respect to the global maximum – considering all time steps – vs. the local maximum – considering each time step separately – has been proven useful [56]. Local scaling emphasizes attribute differences within a network generation, while global scaling is suitable for comparing attributes between different generations. In addition, a node degree histogram is generated for each network generation currently displayed.

For comparing different network generations from a topological point of view, the user may select a number of time steps in the *Dynamic Graph Control Panel* and apply operators on the node and edge sets of the chosen graph to filter certain elements of the super graph. Set operators include *AND, OR*, and *DIFF* for the symmetrical difference between different network snapshots. This is used for detecting subset relations and selecting appearing or disappearing elements (see Figure 4.5b).

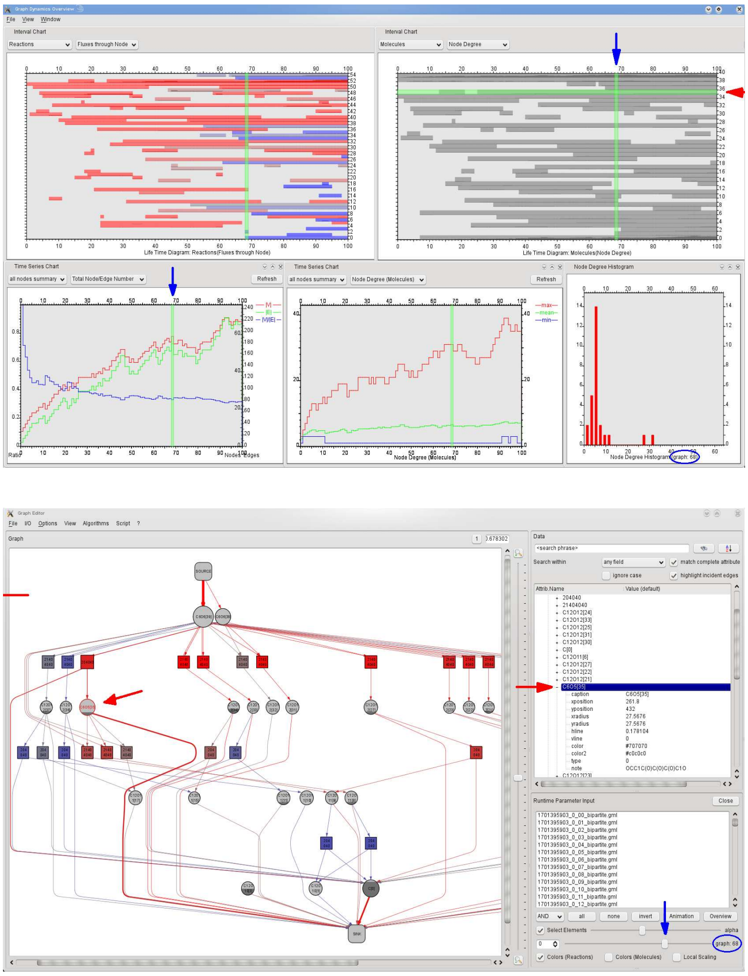
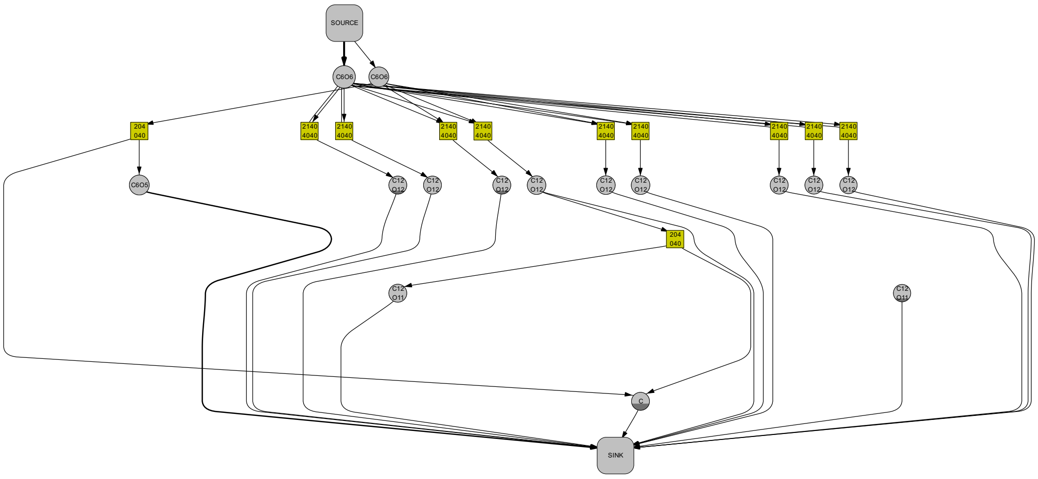
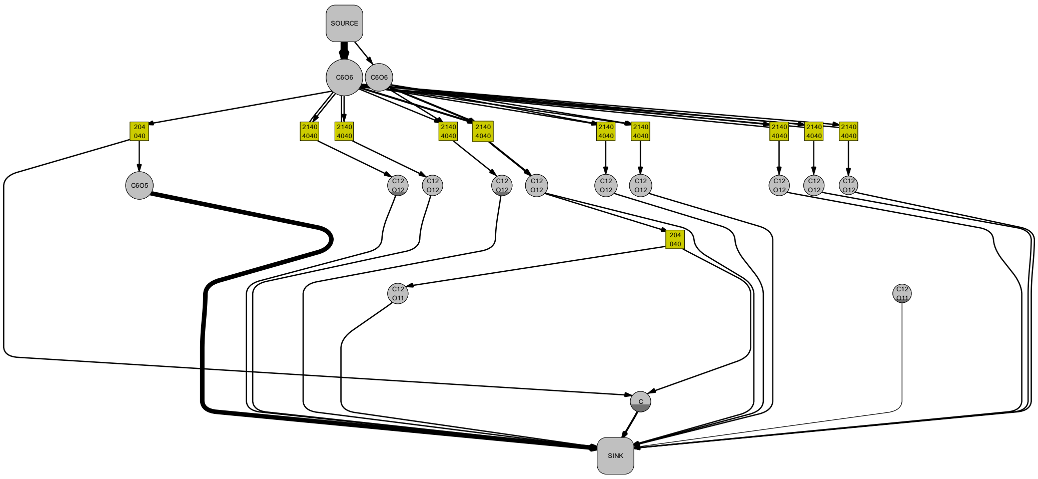


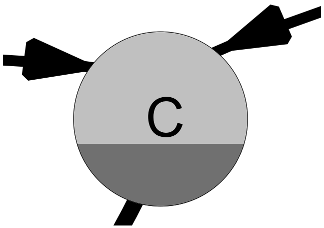
Figure 4.6: Linked View realization facilitates browsing different graph snapshots in time. The blue arrows indicate the current position in time, red arrows indicate the selected node in the current generation. These components of the graphical user interface are also sensitive to user input and can be used for navigation. Selecting a node in the Graph Scene (bottom) highlights the associated row in the appropriate interval chart as well as the associated point in time in all charts. The five diagrams given on the upper display the following data. Top: Life time diagram of reactions overlaid with the number of pathways through each reaction node. Life time diagram of metabolites overlaid with each node's degree. Bottom: Time series chart giving number of nodes, edges, and nodes-to-edges-ratio. Time series chart of summarized node degree (minimum, maximum, average) over all metabolites. Node degree histogram of the currently displayed graph generation.



(a) Global scaling for easy comparison of different points in time.



(b) Local scaling for emphasizing flux differences within one network step.



(c)

Figure 4.7: Global versus local scaling for visualizing graph attributes. Note the dark gray “filling level” in the metabolite nodes depicting the current concentration value (c).

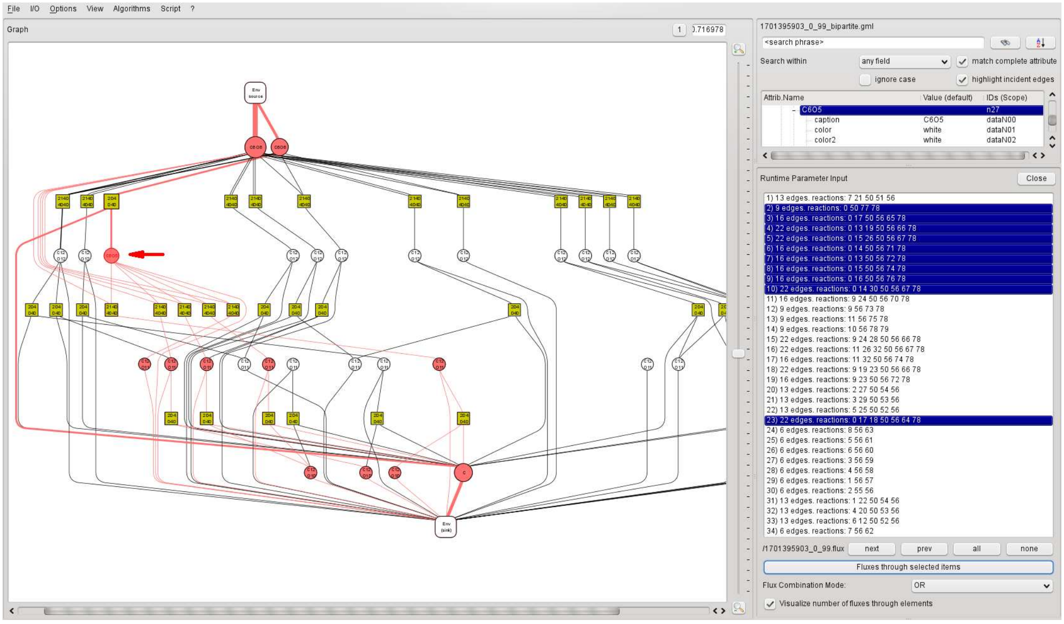
4.4.3 Detailed View

In this section, the user takes a closer look at the emergence of individual elementary pathways (fluxes) in a single network evolution step. The aim is to further investigate elements being more or less likely to participate in pathways through the metabolic network and to identify individual elementary pathways. As described in the previous section, the user has identified reactions and metabolites preferred to form pathways as well as key enzymes with high activity. Interactivity plays a crucial role in this analysis step. There are two methods of operation: First, the user can select any number of elementary pathways to be highlighted in the Graph Scene displaying the current network generation. Second, the previously identified key elements can be selected in the Graph Scene for highlighting all associated elementary pathways. See the screen shot given in Figure 4.8. We again implemented set operators on the selected nodes applied for the flux visualization. We found that this is a highly flexible and intuitive way to detect pathways running through all the selected elements – *AND* operator, at least one of the selected elements – *OR*, or none of the selected elements – *NOT*.

Concerning the attribute dynamics associated with an enzyme, reaction, or metabolite, we take advantage of the linked view implementation depicted in Figure 4.6 to display the attribute development of the selected node over time. Selecting a different node instantly updates the displayed time series of the chosen attribute.

4.5 Results

We will present an excerpt of the analysis of metabolic evolution on one simple example simulation run, illustrating the usefulness of the visualization. The simulation takes two molecules as steady input, namely, the sequential and cyclic form of glucose. The particular enzymes are determined by the random genome, their catalytic activities encompass the full set of chemical reactions. The simulation is run as an adaptive walk over 100 generations. In Figure 4.9, four snapshots from different stages (generations) of the metabolic network evolution are depicted. The information about the life time of all reactions during the entire simulation is kept in Figure 4.10. Some basic properties, such as node degree and number of



Attrib.Name	Value (default)	IDs (Scope)
C6O5		n27
caption	C6O5	dataN00
color	white	dataN01
color2	white	dataN02

```
1) 13 edges. reactions: 7 21 50 51 56
2) 9 edges. reactions: 0 50 77 78
3) 16 edges. reactions: 0 17 50 56 65 78
4) 22 edges. reactions: 0 13 19 50 56 66 78
5) 22 edges. reactions: 0 15 26 50 56 67 78
6) 16 edges. reactions: 0 14 50 56 71 78
7) 16 edges. reactions: 0 13 50 56 72 78
8) 16 edges. reactions: 0 15 50 56 74 78
9) 16 edges. reactions: 0 16 50 56 76 78
10) 22 edges. reactions: 0 14 30 50 56 67 78
11) 16 edges. reactions: 9 24 50 56 70 78
12) 9 edges. reactions: 9 56 73 78
13) 9 edges. reactions: 11 56 75 78
14) 9 edges. reactions: 10 56 78 79
15) 22 edges. reactions: 9 24 28 50 56 66 78
16) 22 edges. reactions: 11 26 32 50 56 67 78
17) 16 edges. reactions: 11 32 50 56 74 78
18) 22 edges. reactions: 9 19 23 50 56 66 78
19) 16 edges. reactions: 9 23 50 56 72 78
20) 13 edges. reactions: 2 27 50 54 56
21) 13 edges. reactions: 3 29 50 53 56
22) 13 edges. reactions: 5 25 50 52 56
23) 22 edges. reactions: 0 17 18 50 56 64 78
24) 6 edges. reactions: 8 56 63
25) 6 edges. reactions: 5 56 61
26) 6 edges. reactions: 6 56 60
27) 6 edges. reactions: 3 56 59
28) 6 edges. reactions: 4 56 58
29) 6 edges. reactions: 1 56 57
30) 6 edges. reactions: 2 55 56
31) 13 edges. reactions: 1 22 50 54 56
32) 13 edges. reactions: 4 20 50 53 56
33) 13 edges. reactions: 6 12 50 52 56
34) 6 edges. reactions: 7 56 62
```

/1701395903_0_99.flux

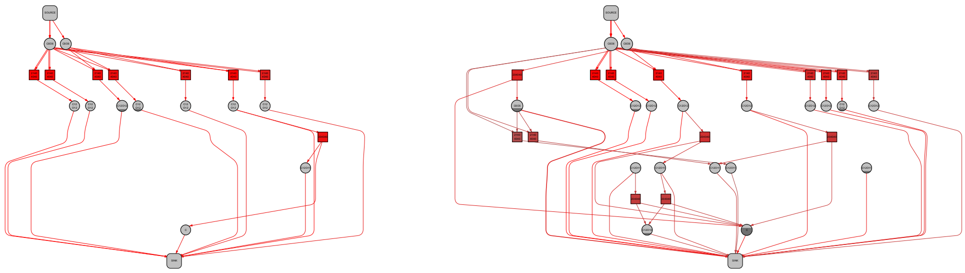
Flux Combination Mode: OR

☑ Visualize number of fluxes through elements

Figure 4.8: Details on Demand: Interactive flux analysis for one chosen time step (here: t=99). Individual elementary pathways can be selected for visualization. All pathways containing the molecule C_6O_5 are highlighted.

reactions and metabolites are recorded in Figure 4.11.

In this analysis we wanted to investigate the early steps in the formation and evolution of metabolic pathways and interpret our findings in terms of existing evolutionary scenarios. We will focus on three popular theories, that can be compared nicely to our results. One of the first theories proposed on this matter is backward or retrograde evolution [33], stating that pathways evolved upwards, in the need of finding more and more distant metabolites to build a particular beneficial substrate due to depletion of metabolites. Contrary, forward evolution [14] suggests the opposite direction of pathway evolution. Due to ever further processing of molecules to gain more beneficial molecules and production of energy, pathways evolve in such a way that ancient enzymes are upstream along the pathway, while younger enzymes are further downstream. For backward evolution we see the opposite picture. The third scenario is the patchwork model [35], which explains the formation of new metabolic pathways through recruiting of enzymes from already existing pathways. Looking at the enzyme age distribution along a pathway or network, one would expect a mosaic-like pattern (see Figure 4.9c).

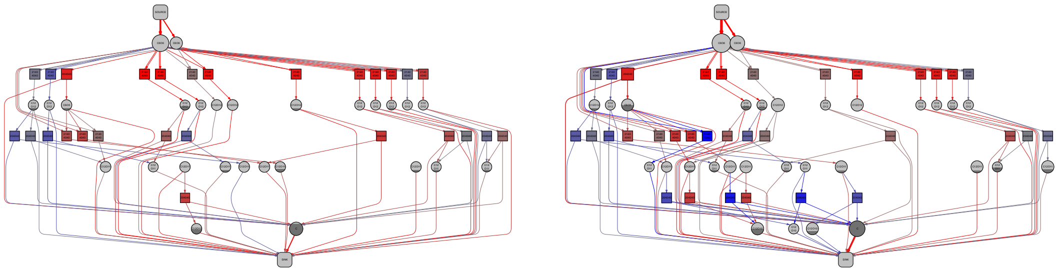


metabolites=11, reactions=8

(a)

metabolites=18, reactions=15

(b)



metabolites=27, reactions=27

(c)

metabolites=30, reactions=30

(d)

Figure 4.9: A series of simulated metabolic networks after (a) 11, (b) 31, (c) 67, and (d) 100 generations. The squares represent reactions. The color of these nodes and their incident edges indicates the first occurrence of the reaction (red: early, blue: late). Gray circles represent metabolites. An edge leading from a metabolite to an enzyme indicates that the respective metabolite is an educt in the reaction. An edge from an enzyme to a metabolite marks it as a product. The size of the nodes and the width of the edges encode for the number of minimal pathways in which the respective object is involved.

The four snapshots in Figure 4.9 showing the metabolic network in different stages are aligned to the Union Graph over all generations. Thus, we can see that in the first steps the reactions upstream in the network are added. The pathways are formed further in this forward direction. Looking at the last generation, basically all pathways from source to sink follow the forward evolution scenario, with older (red) enzymes being at the top (upstream) and younger (blue) enzymes more at the bottom (downstream). This observation is further established through the interval graph for all chemical reactions in Figure 4.10. The reactions are here ordered according to their position in the graph. There is a clear trend of older reactions being on the top and younger ones following more downstream. If we compare the colored bars (Figure 4.10c) showing the enzyme age distribution for our results and the three scenarios mentioned above, the pathway evolution again seems to explain our results best. Therefore, it appears that in the early phase of metabolic evolution, forward evolution is dominant. In a recent study [75], we tested this speculation with a more exhaustive approach using 100 simulations with more complex settings and metabolic networks. Similar to this analysis, we find forward evolution to be acting in the early steps of pathway formation. In later stages, enzyme recruitment seems to take over. However, a core of forward evolved pathways from the beginning seems to remain.

We turn now to the evolution of general properties of the metabolic networks from our simulation. The numbers of metabolites and chemical reactions (see Figure 4.11a) develop with almost the same rate. This indicates that most metabolites are only involved in exactly one reaction. Combining this reasoning with the observation that the maximal node degree of metabolites increases significantly more than their average node degree (see Figure 4.11b), we can conclude that our metabolic networks evolved one or only a few highly connected metabolites, socalled hub-metabolites, and probably has a scale-free node-degree distribution, typical for real-world metabolic networks. Another observation is the steady increase of the average enzyme connectivity while the average metabolite connectivity converges. The explanation for the latter is the high number of metabolites involved in only one reaction. A similar trend will likely arise in more complex

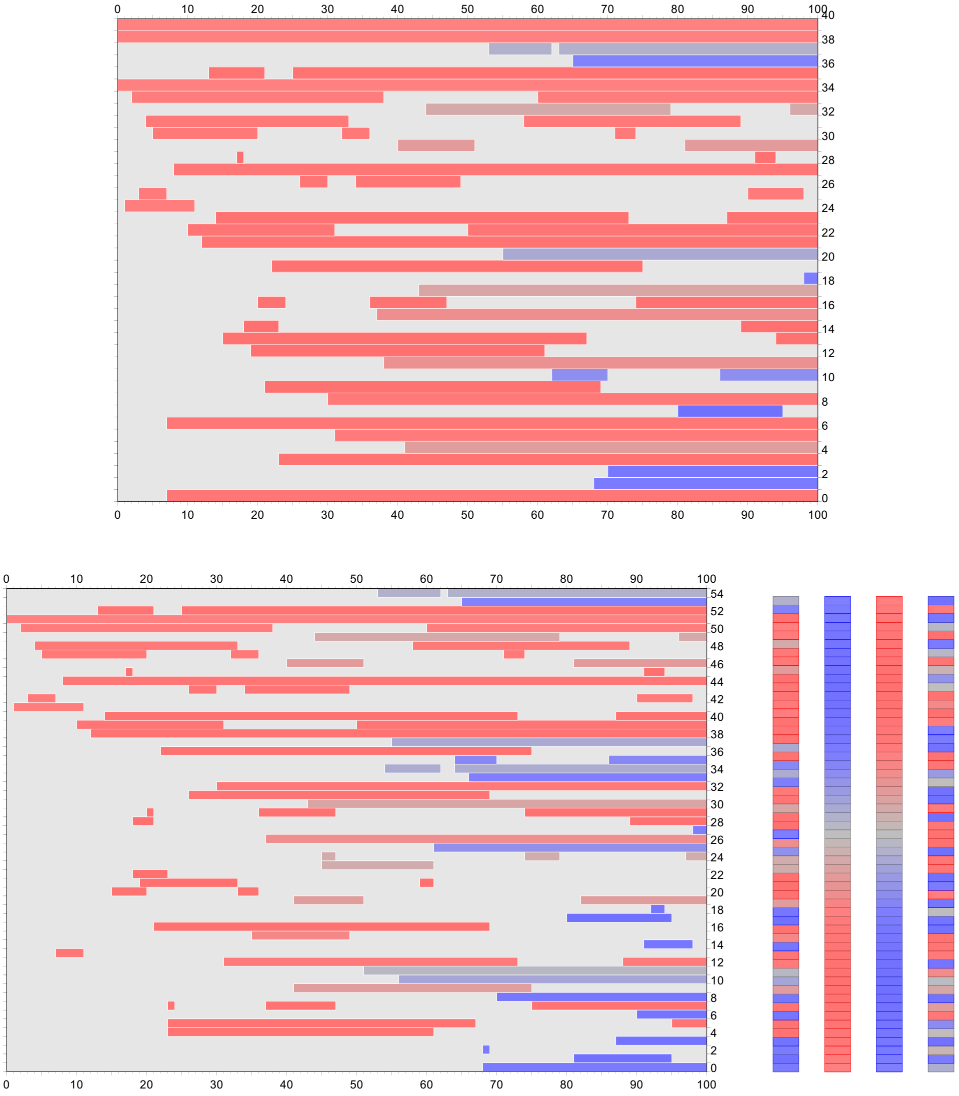
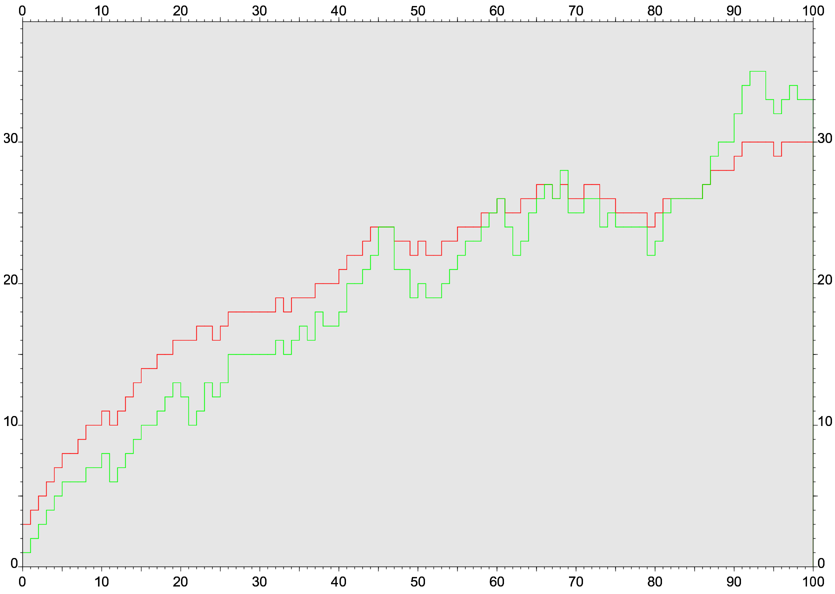
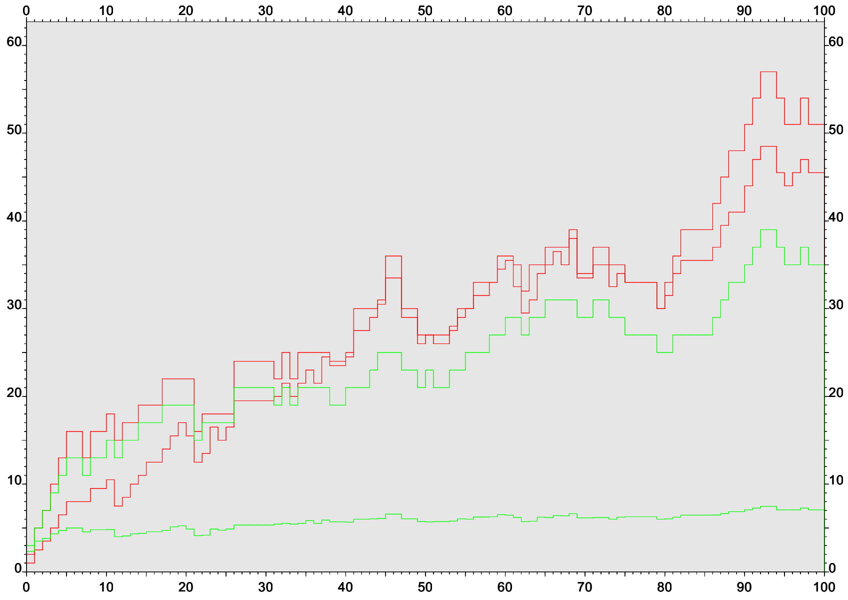


Figure 4.10: Overview visualization of the analyzed dataset. Life time diagram of metabolites (top) and reactions (bottom). Their position in the diagram (y-axis) reflects the associated nodes' positions in the graph layout. The DOT layout algorithm places reaction nodes close to the source metabolites in upper positions, and reaction nodes close to the excreted metabolites at the bottom. The color scheme (bottom, right) given on the first vertical bar lists the node colors as they appear from top to bottom and from left to right. The second bar depicts idealized retrograde evolution, the third idealized forward evolution, and the last bar with random coloring depicts an instance of the patchwork model, respectively. Comparing our scenario with the three evolution models, our data bear resemblance to the forward evolution model.



(a)



(b)

Figure 4.11: Tracking selected attributes over time. (a) Number of metabolites (green) and reaction nodes (red). (b) Node degree (maximum and average) of metabolites (green) and enzyme nodes (red).

stages for enzyme connectivity as well.

4.6 Conclusion

In this chapter we have presented an extension to our existing graph visualization system to support the exploration and analysis of dynamic metabolic networks. The development process was intensively accompanied by the scientists providing the data and was found to be extremely helpful to understand the underlying mechanisms of metabolic network and biochemical pathway evolution. The visualization could reveal general properties of the considered systems in terms of network topology, but also answered specific questions on the evolution of metabolic networks and the emergence of pathways within the network. The results from the visual analysis stated in this work could be confirmed and validated by statistical methods presented in [75].

We found that interactivity plays a crucial role in the analysis process. It was successfully implemented using linked views for fast and intuitive navigation in time as well as within a selected network configuration. We intend to examine more simulation runs with different parameter configurations to compare the results and to gain a deeper understanding of metabolic network evolution.

For laying out the constructed Super Graph, the Sugiyama method has proven to produce the best results. The layout algorithm was a suitable choice due to the fact that the considered network contained only a few number of cycles, and therefore, the observed elementary pathways followed the general direction from top (source nodes) to bottom (sink nodes). The major disadvantage of this layout method is the amount of space required for the drawing. The number of graph elements in the Super Graph was small enough for a feasible application of this layout algorithm. Datasets with more generations can become very large and too complex for using the applied graph layout. However, there is room for improvement, since many elements in the Super Graph do not overlap in time and may therefore occupy the same position reducing the total space for the layout. The same argument holds for simultaneously laying out the described reaction and enzyme view as the set of reaction nodes and the set of enzyme nodes are disjoint. The improved layout method for dynamic graphs taking advantage of that fact is presented in [17].

5 Comparative Visualization of Network Graphs

The third aspect of visualizing biochemical content we want to focus on is the investigation of how graphs representing metabolic data can be compared. This work is application driven and shall answer two fundamental questions:

- How does the metabolism differs from one species to another?
- How can metabolic network data from one resource be compared to a different resource?

The first question is of course biologically motivated. As we have discussed in the previous chapter, metabolic networks underlie evolutionary mechanisms leading to the assumption that different organisms may have realized metabolic processes in different ways. While a specific chemical reaction can be considered as a constant unit in the network defined by its stoichiometrics, i.e., the proportion of substrates and products of a chemical reaction, reactions do not take place in every environment due to thermodynamic constraints. More specifically, the activation energy is responsible for how fast substrates are converted into products and may prevent the conversion under normal circumstances. The catalyzing function of enzymes is the reduction of the activation energy increasing the conversion rate, however enzymes often depend on a specific range of temperature and pH value. Furthermore, some environments may even be toxic to most of the organisms while others dwell under those conditions. The adaption to different environments raises two issues we want to discuss in the following section: Which metabolic pathways are realized in different organisms and if there is a specific pathway present in two species, how do these differ from another in terms of chemical reaction sets.

The second question mentioned above is of a rather technical nature. Many biological databases make metabolomic data available, however the organisation of these data, i.e., classification of reaction sets into metabolic pathways as functional subunits, may differ as well as the coverage. Another aspect to consider is the correctness of data provided by the respective resources. We will address these issues in section 5.2.

5.1 Inter-species Comparison of KEGG Pathway Networks

Many metabolic pathway resources offer reaction networks from different organisms. While the metabolism of a few model organisms, e.g., Saccharomyces cerevisiae (yeast) and Drosophila melanogaster (fruit fly) as eukaryotic organisms, or Escherichia coli as procaryotic organism is fairly well understood and widely available, the KEGG Organism branch of the KEGG Pathway database provides metabolic networks of as much as 1371 studied species in addition to the 140 reference pathways examined in chapter 3. A fundamental question in this context is, to what extend corresponding pathways in different organisms are identical or, if present, where different reactions realize the same metabolic function. Two scenarios can be considered: The comparison of analogue metabolic pathways in two different organisms, and the comparison of organism-specific pathways against the reference pathways.

The KEGG Pathway web interface does not provide a way for comparing parts of the metabolism between different species. However, the set of organism-specific reactions can be displayed by highlighting the particular chemical reactions in the reference map for one selected pathway. The reactions of organism-specific pathways are depicted as green boxes, which are hyperlinked to GENES entries, indicating the presence of genes in the genome and also the completeness of the pathway [1].

We follow a more global approach and adhere again to Ben Shneiderman's information visualization paradigm. For all organisms found in KEGG Organism, a compact table view depicts the set of metabolic pathways being present in a particular organism and informs about the sets of identical pathway realizations. Based on the layout of the reference map, the chemical reaction networks of two selected organisms can then be visually compared by applying set operators on the vertex and edge sets.

5.1.1 Organism Overview Visualization

The KEGG ORGANISM database comprises 140 metabolic pathways, of which only a subset is present in the 1371 species. In our analysis, we included 94964 organism-specific pathways stored in KGML files. These were converted to graphs in the same way as described in section 3.1.2. Given the set of organisms $O, \mid O \mid = 1371$ and the set of pathways $P, \mid P \mid = 140$, we can define the set of pathway realizations $R \subseteq O \times P$ with the organism-specific pathway realization $r_{o,p} \in R$. In our case, $\mid R \mid = 95118$. In the overview exploration phase of the analysis, we are interested in two sets:

- The organism-specific set of pathways:
  $P(o), o \in O, P(o) \subset P$.
- The set of organisms realizing a particular pathway:
  $O(p), p \in P, O(p) \subset O$.

Due to the fact, that many organisms realize a pathway in exactly the same way, i.e., many pathway realizations are equivalent in terms of reactions and chemical compounds, the above number of pathway realizations greatly reduces to a small number of equivalence classes: $[r_p]_\sim \subseteq R(p)$ defining a partition on the organisms having a particular pathway $O(p)$.

The image in 5.1 visualizes the sets $P(o)$ and $O(p)$ in a compact boolean matrix. The implementation of our analysis tool allows interaction with the graphical output in two ways:

1. The columns can be reordered with respect to a selected row (pathway) to emphasize the set of organisms having that particular pathway, $O(p)$.
2. A column (organism) can be selected to highlight all pathways present in that organism, $P(o)$, and to highlight the equivalence class containing that organism-specific pathway in all of the partitions.

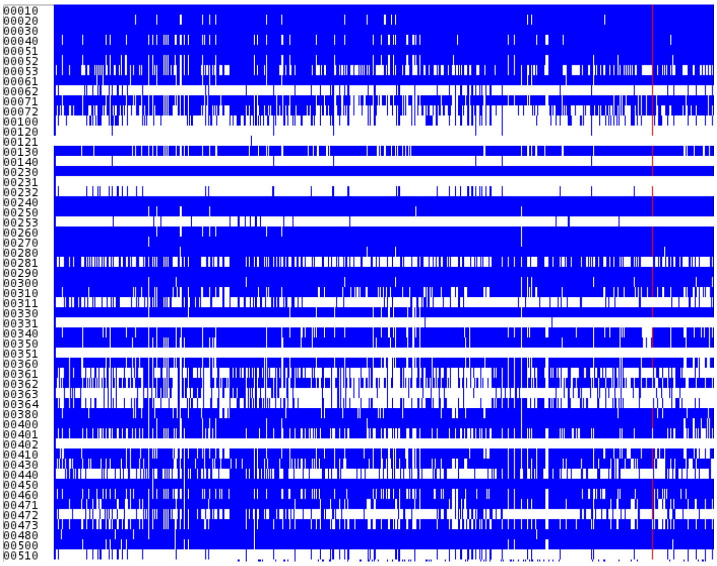


Figure 5.1: Pathway realizations matrix for the first 55 pathways (rows) and the first 650 organisms (columns) according to KEGG. Each entry indicates, if a particular pathway is present in an organism (blue) or not (white). The column for *hsa* (Homo sapiens) is highlighted in red.

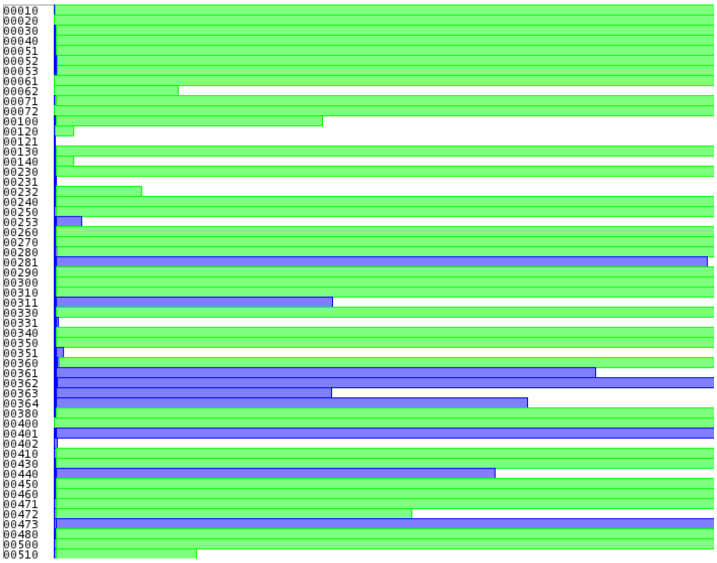


Figure 5.2: Partitions for the first 55 pathways. Each row depicts the pathway-specific partition on the organism set. The width of the rectangles is determined by the number of organisms of the respective equivalence class. All sets containing the pathway realized in *hsa* (Homo sapiens) is highlighted in green.

The image in 5.2 displays all possible partitions on the organism sets defined by the pathways. Each partition is depicted as a bar subdivided into the partition's equivalence classes. The size of each subdivision reflects the size of the equivalence class, i.e., the number of organisms realizing that particular pathway.

5.1.2 Chemical Network Comparison

After selecting a set of pathways of interest for two organisms a union graph containing all elements of the different pathway realizations is constructed. Considering the two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ for the pathway realizations of the first and second selected species, the union graph is defined as $G = (V_1 \cup V_2, E_1 \cup E_2)$. The union graph computation is very straightforward, since all reactions and chemical compounds of the considered metabolic networks have unique identifiers provided by KEGG, which are consistent among all organism-specific pathway graphs. These IDs are used to map the nodes of G_1 to the corresponding nodes of G_2. In addition to the connectivity, G also preserves the 2-layer hierarchy of G_1 and G_2 with pathway nodes on the top level containing the chemical networks as nested graphs. We can explore the graph in the same manner as suggested in chapter 3 from top to bottom. Set operations applied to nodes and edges of the individual pathway realizations select and highlight elements in the following ways:

- G_1 only
- G_2 only
- $G_1 \setminus G_2$
- $G_2 \setminus G_1$
- $G_1 \cap G_2$
- $G \setminus (G_1 \cap G_2)$ (symmetrical difference)

The exploration process starts with a fully collapsed network, i.e., only the top-level pathway nodes are visible. If elements of the bottom layer (reaction and compound nodes) are selected in the manner described above, the parent nodes of

ID	Pathway name	Reference Pwy		H. sapiens (hsa)	
		Cpd	Rct	Cpd	Rct
00730	Thiamine metabolism	26	23	26	20
00740	Riboflavin metabolism	21	17	21	14
00750	Vitamin B6 metabolism	32	40	32	19
00760	Nicotinate and nicotinamide met.	44	57	44	29*
00770	Pantothenate and CoA biosynthesis	27	30	27	23*
00780	Biotin metabolism	11	10	11	9

Table 5.1: List of pathways depicted in Figure 5.3. Except the two pathways *00760* and *00770*, the *hsa* realizations are subgraphs of the reference pathway. In the two cases (*), there is one additional node referencing a gene product only present in Homo sapiens. These nodes are isolated, i.e., not connected to any chemical compound node.

the selected elements are also highlighted (see Figure 5.3, box) giving an overview on differences or similarities in the underlying chemical networks. We can then expand the highlighted pathway nodes to explore the detailed network (Figure 5.3, large image). The two images show the graph, which was constructed from six metabolic pathways belonging to the metabolism of cofactors and vitamins (see Table 5.1).

5.1.3 Results

The analysis of a total of 94964 pathways for 1371 organisms and the set of reference pathways reveals several interesting facts about the data provided by KEGG ORGANISM. Table 8 in the Appendix gives an overview on the identified equivalence classes for every pathway. Many pathway realizations are indeed identical. The detailed analysis described in the previous section exposes a large number of isolated nodes representing gene products, which are specific to the respective organism. Taking isolated nodes into account naturally results in a higher number of equivalence classes. However, these do not contribute to the metabolic network information of the graph. The removal of isolated nodes representing gene products prior computing the equivalence classes on the pathway realizations is based on this rationale. The next-to-last column of Table 8 depicts the number of different pathway realizations being considerable smaller than the number of organisms re-

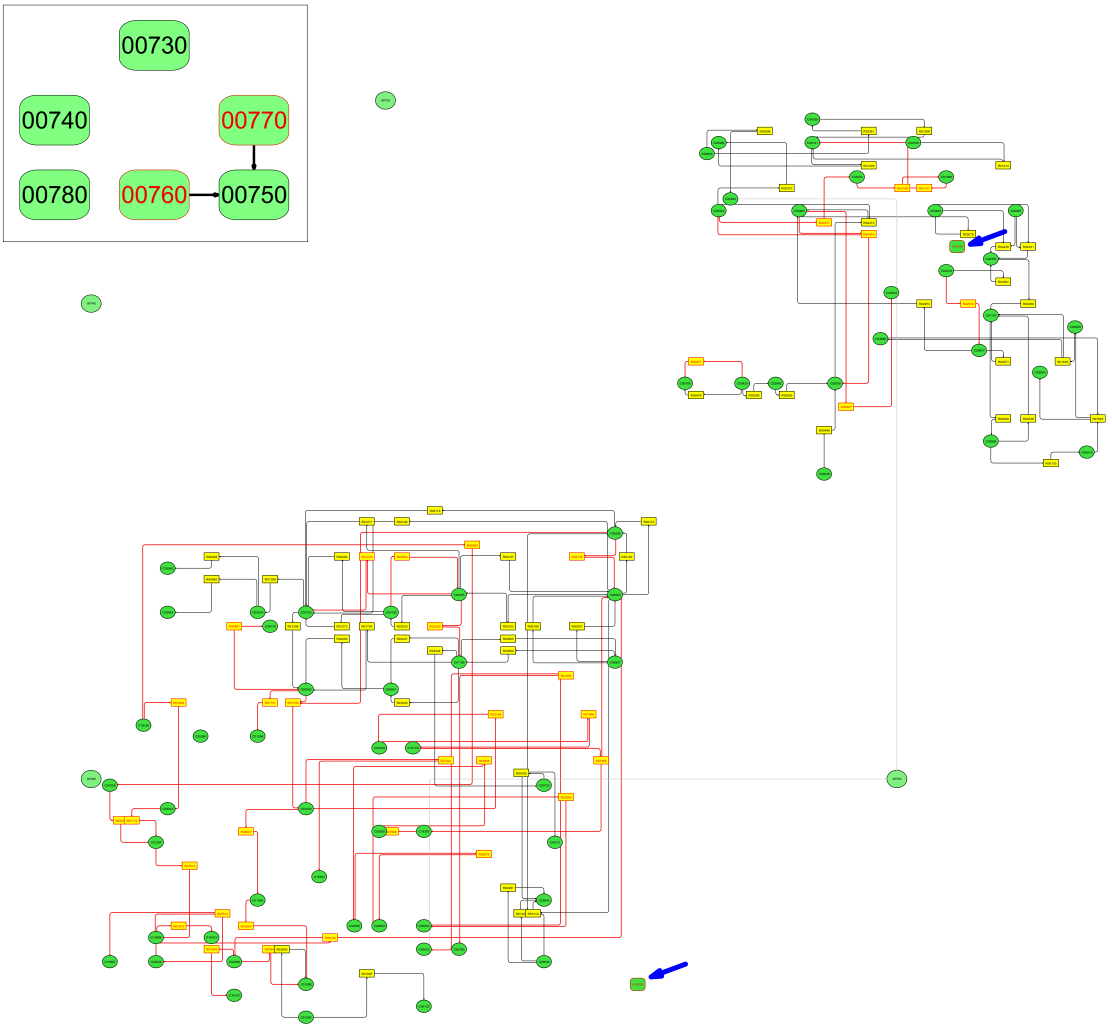


Figure 5.3: Top-Level graph of selected pathways (reference map vs. homo sapiens *hsa*) and detailed view on two expanded pathways. The highlighted pathway nodes (box, top-left) indicate that there are elements in the *hsa* realizations which are not present in the reference pathways ($G_2 \setminus G_1$). The highlighted elements in the detailed view of pathways *00760* and *00770* represents the symmetrical difference $(G_1 \cup G_2) \setminus (G1 \cap G2)$. The blue arrows point to the two nodes present only in the *hsa* realizations.

alizing this pathway. Removing isolated nodes representing gene products reduces this number even further as depicted in the last column of Table 8. In fact, most of the metabolic pathways are realized in *exactly the same way* in every organism observed.

By not taking isolated elements into account, another observation can be made. Almost all metabolic pathways realized in different organisms are subsets, or subgraphs, of the reference pathway. As Figure 5.3 suggests, the subgraph may be considerably smaller than the reference pathway graph. This is at least questionable and should be veryfied by experts in the field of biochemistry, as the deviation from the reference pathway together with the lack of deviation in pathway realizations among different organisms may point to artifacts or missing knowledge in the metabolic network data provided by KEGG ORGANISM. The next section will address this issue by comparing metabolic network data of KEGG PATHWAY to a different resource.

5.2 Visual Comparison of KEGG Pathway and BioCyc content

Bioinformatics research in general and the exploration of metabolic networks in particular rely on processing data from different sources. Visualization in this context supports the exploration process and helps to evaluate the data quality of the used sources.

In this work, we extended our existing metabolic network visualization toolbox and hereby address the fundamental task of comparing metabolic networks from two major bioinformatics resources for the purpose of data validation and verification. This is done on different levels of granularity by providing an overview on retrieval rates of chemical compounds and reactions per pathway on the one hand, as well as giving a detailed insight into the differences in the biochemical reaction networks on the other.

We reconstructed different subsets of the metabolism stored at the KEGG Pathway database and compare these networks against the complete metabolic network provided by the MetaCyc branch of the BioCyc database collection (`http://biocyc.org/`). Matches among the sets of chemical compounds and reactions are highlighted and propagated to higher levels of abstraction to infere pathway correspondence between the two resources.

5.2.1 Motivation

During the last decade a wealth of high throughput sequence data of both genomes and proteomes have become available for a wide variety of organisms. For a small number of model organisms, on the other hand, detailed information is available on their metabolic chemical reactions and the enzymes that catalyze them. Combining these sources of knowledge allows the inference of metabolic networks, see e.g. [40]. Databases, including KEGG Pathway [39, 37], BioCyc [13, 12], WikiPathways [59], Reactome [31] as well as a wide variety of species-specific resources such as LeishCyc [18] provide metabolic network data with a varying degree of manual curation.

Both the computational inference of metabolic networks and the manual curation process is subject to errors, however. Systematic misannotations and ambiguities in assignments of enzyme functions [29, 65], for instance have been identified as sources of errors. The most common type of error is associated with "over-prediction" of molecular function. Further problems arise from the incomplete modeling of the chemical reactions themselves, which typically are treated as annotation texts rather than data in their own right [57] and can lead to stoichiometric inconsistencies [26] that can hamper the analysis of metabolic data. The notorious incompleteness of genome annotations even in well-studied organisms such as *E. coli*, yeast, or human, furthermore translates into an incompleteness of metabolic pathway maps. Enzymes also may change their function and substrate specificity over the course of evolution, fundamentally limiting the accuracy of functional annotations that are rooted in sequence similarities. Taken together, thus even well-curated metabolic network data cannot be assumed to be complete and entirely accurate.

The direct comparative analysis of the chemical reaction networks describing the metabolism can be used to identify and expose potential weaknesses and errors in the representation of biochemical networks. Our approach is inspired by a set-theoretic approach to comparing chemical (and in particular metabolic) networks [22], originally proposed as a means of identifying metabolic innovations.

In this contribution we focus in particular on the KEGG PATHWAY Database. KEGG pathways are relatively large sub-systems of the metabolic network that combine multiple biological processes from different organisms in a way that matches biological intuition, but lacks a formal definition in terms of the underlying reaction network [28]. The database contains a set of manually drawn metabolic pathway diagrams presented as semi-static visualizations used for navigating the data on line as well as XML-like descriptions of those pathways. We use this network as a template graph and perform a multi-scale comparison to the metabolic network provided by BIOCYC. Firstly, we describe how compounds and reactions are matched between the two networks and how this information is used to infere relationships between pathways defined by KEGG vs. the ontology defined in BIOCYC. The results can be viewed by the user from a global point of view, i.e., a quantification of the node matching quality for each pathway of the KEGG network, and in more detail by interactively expanding the pathways of interest to reveal the respective reaction networks. Brushing techniques for highlighting portions of the detailed network are used to draw the users attention to inconsistencies or ambiguities among the two resources. We combine several popular information visualization methods to navigate the presented network, such as semantic zoom, hierarchical exploration by node expansion, and focus&context techniques. Once the user has identified network points of interest based on the highlighted differences, we provide the context of the matched reaction or compound in the BioCyc graph as overlay on the current KEGG network. Additionally, the respective subset of BioCyc's pathway ontology can be viewed on demand for every reaction, compound, and pathway node. The implementation of the exploration process is inspired by Ben Shneiderman's mantra of visual information-seeking [69].

5.2.2 Data Resources, Preprocessing, and Data Structure

Both databases provide a semi-structured flat-file dump of metabolic network data, either as KGML files (KEGG) or attribute-value files (BioCyc). The reconstruction of the chemical reaction networks from KGML files is a straight-forward task and was already explained in chapter 3. Each KGML file represents a metabolic pathway as defined by KEGG and contains the connectivity information of reaction and compound nodes, as well as layout information for each node. This allows a very similar depiction of the pathway maps as provided by the KEGG system. It helps preserving the mental map as these drawings constitute a de-facto standard among biologists. After the construction of bipartite graphs representing the pathways, we add parent nodes for each pathway and insert inter-pathway edges connecting two identical compounds in different pathways as defined by the *maplink* elements in the KGML file. These inter-pathway edges are propagated to the higher pathway level and will be visible in the abstracted network overview part of the visualization. The hierarchy introduced by the pathway nodes is non-overlapping due to the duplication of compound nodes present in more than one pathway. Within a pathway, all compound and reaction nodes are unified facilitating the mapping process from one network to the other.

In the case of BioCyc, we only use the MetaCyc branch of the database collection containing multi-organism metabolic pathways. It relates most closely to the reference pathways from KEGG as these also represent the union of the reaction sets realized in different organisms. Unlike the KEGG Pathway graph, we construct a large unified bipartite graph from a reactions data file. As for the pathway ontology, we add nodes for each occurring pathway and evaluate the super-pathway and sub-pathway relations to reconstruct the ontology represented as a directed acyclic graph. Each pathway references several reactions, creating an overlapping hierarchical clustering of the reaction set. The annotations, i.e., synonym lists for chemical compounds and enzyme commission numbers (EC) for reactions are stored and serve as the basis for the mapping process.

5.2.3 Graph Matching

We use the metabolic network constructed from the KEGG Pathway database as template graph, which is considerably smaller than the BioCyc graph, but not necessarily a subset. The mapping we describe here is uni-directional from KEGG to BioCyc, so we identify three cases:

1. Unique match: a node in the KEGG network can be mapped to exactly one node in the BioCyc network.
2. Ambiguous match: a KEGG node will be mapped to more than one BioCyc node.
3. "No hit": the KEGG compound or reaction cannot be found in the BioCyc collection.

The actual mapping process is done in two steps: (1) matching compound nodes, and (2) match reaction nodes based on the compound mapping.
Given two graphs, finding a graph isomorphism or inclusion relations is NP-hard. However, the problem described here is not of a graph-theoretical nature in the traditional sense, but rather a lexicographical one. With every chemical entity comes a set of annotations from the respective database. For mapping the chemical compounds found in the KEGG network to nodes in BioCyc, we use a list of synonymous chemical names associated with the compound. These two sets of lists will be cross-referenced to identify matches. For every compound node in the template network (KEGG), we hold a – possibly empty – list of matching candidates in the BioCyc network.

In the second step, we do not have to rely on string comparison operations. Instead, we take advantage of the unique adjacency of a reaction node. In general, a reaction is defined by the sets of chemical compounds it consumes – substrates – and the set of compounds it produces – products. Given this signature, we can robustly identify nodes in the bipartite BioCyc network that fulfill a certain neighborhood configuration. For each reaction node in the KEGG graph, we determine the set of adjacent compounds and identify the set of reaction nodes in the *BioCyc* graph that have the matched compounds as neighbors. In case of ambiguous compound matches, we have to repeat the search for every combination of potential compound candidates, resulting in a potentially larger set of reaction matches. Reactions can still be robustly identified even if one or more compounds in its neighborhood could not be matched at all. We refer the reader to section 5.2.5 for a detailed discussion on the different mapping scenarios. For multiple hits on reaction nodes, we use the EC nomenclature given with the reactions to refine the search result.

As a measure for the overall quality of the mapping serves a simple match score s_p for each pathway p in KEGG, which accumulates the match score s_v for each node $v \in V_p$:

$$s_v = \begin{cases} 0 & , \mid m(v) \mid = 0 \\ \frac{1}{|m(v)|} & , \mid m(v) \mid > 0 \end{cases} \text{ and } s_p = \frac{\sum_v s_v}{|V_p|}$$

It takes ambiguous matches $m(v)$ into account and penalizes a large number of candidates for a specific node.

5.2.4 Visual Comparison and Exploration

After the mapping process has been completed, the overview on the metabolic network comparison is presented to the user (see Figure 5.4). We start with a completely collapsed network with only the pathway nodes visible. Several properties of the underlying networks will be visualized: relative pathway size, match score s_p as the green "filling level" and the number of entities without match in relation to the network size (saturation of the red color).

For a selected pathway node, the user may investigate the relations to the BioCyc pathway ontology (Figure 5.5). On demand, the portion of the directed acyclic

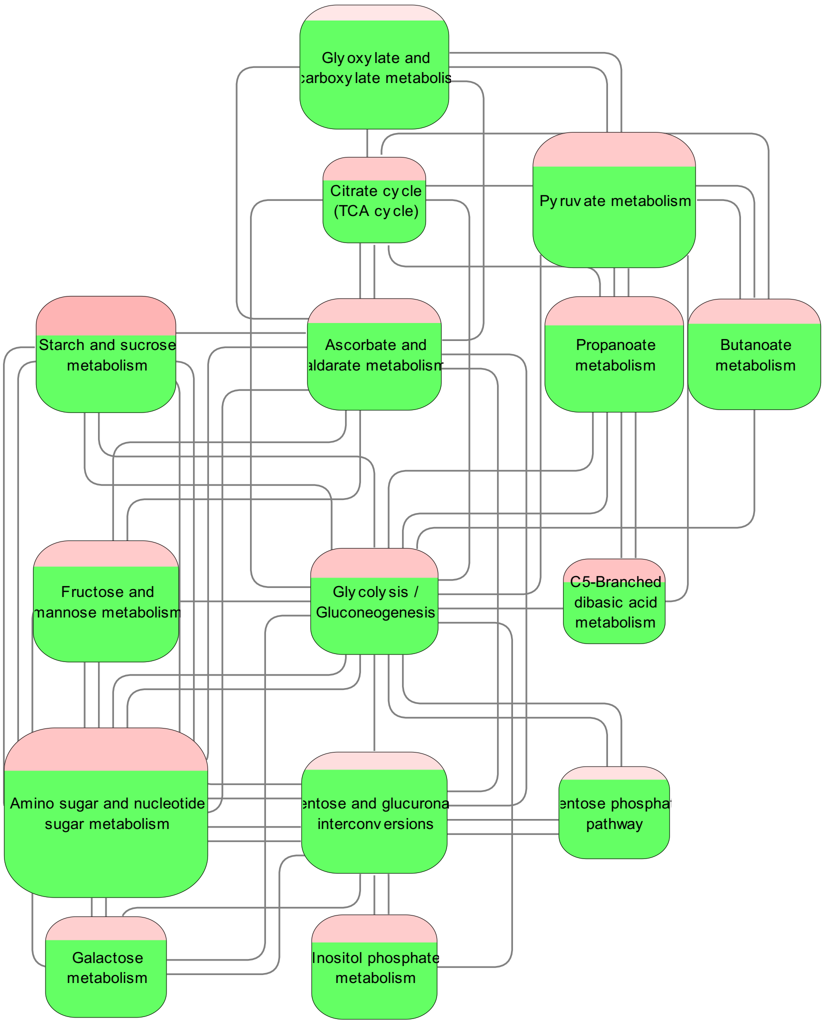


Figure 5.4: Overview on the metabolic network constructed from KEGG PATHWAY. 15 Pathways associated with the carbohydrate metabolism are shown. The node size depicts the number of reactions and compounds of the respective pathway. The nodes' filling level reflects the match score s_p for the respective pathway, which is closely related to the ratio of matched nodes and total node number, but also penalizes ambiguous matches. The saturation of the red color in the upper portion of a node hints to the relative number of nodes that could not be matched at all.

graph representing the nesting relations of pathways together with the respective chemical reaction nodes is overlaid onto the current view. The selected KEGG pathway node remains highlighted. The subgraph of the DAG contains at least all the matched reactions belonging to the selected KEGG pathway. Reactions present in the current subset of the ontology but not part of the selected KEGG pathway are not displayed to avoid clutter.

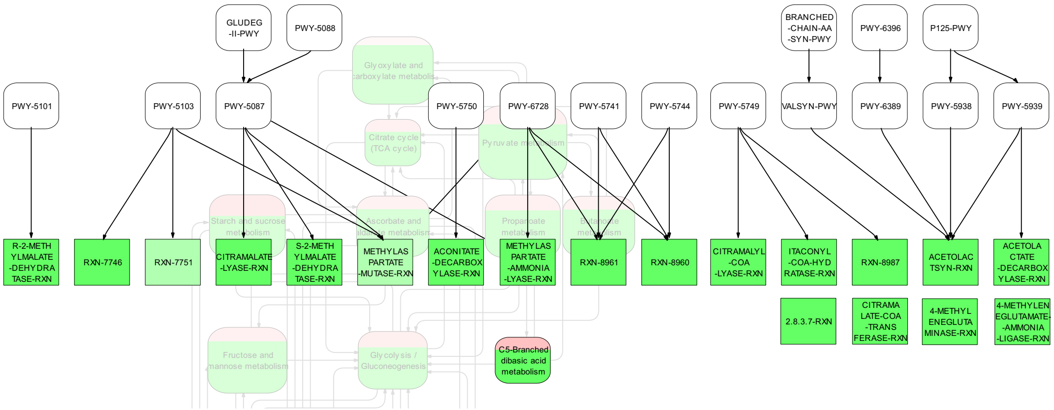

GLUDEG -II-PWY
PWY-5088
BRANCHED -CHAIN-AA -SYN-PWY
PWY-6396
P125-PWY
Glyoxylate and carboxylate metabolis
PWY-5101
PWY-5103
PWY-5087
PWY-5750
PWY-6728
PWY-5741
PWY-5744
PWY-5749
VALSYN-PWY
PWY-6389
PWY-5938
PWY-5939
Citrate cycle (TCA cycle)
Pyruvate metabolism
Starch and sucrose metabolism
Ascorbate and aldarate metabolism
Propanoate metabolism
Butanoate metabolism
R-2-METH YLMALATE -DEHYDRA TASE-RXN
RXN-7746
RXN-7751
CITRAMALATE -LYASE-RXN
S-2-METH YLMALATE -DEHYDRA TASE-RXN
METHYLAS PARTATE -MUTASE-RXN
ACONITATE -DECARBOX YLASE-RXN
METHYLAS PARTATE -AMMONIA -LYASE-RXN
RXN-8961
RXN-8960
CITRAMALYL -COA -LYASE-RXN
ITACONYL -COA-HYD RATASE-RXN
RXN-8987
ACETOLAC TSYN-RXN
ACETOLA CTATE -DECARBOX YLASE-RXN
2.8.3.7-RXN
CITRAMA LATE-COA -TRANS FERASE-RXN
4-METHYL ENEGLUTA MINASE-RXN
4-METHYLEN EGLUTAMATE- -AMMONIA -LIGASE-RXN
Fructose and mannose metabolism
Glycolysis / Gluconeogenesis
C5-Branched dibasic acid metabolism

Figure 5.5: Visualization of pathway relations from a selected KEGG pathway node (C5-branched dibasic acid metabolism) to the pathway ontology provided by *BioCyc*.

Once the user selects a pathway for further investigation by expanding one or more pathway nodes, the detailed chemical network as found in KEGG PATHWAY is revealed (see Figure 5.6). The three types of matches are indicated by different node colors. The white color is used for compounds unable to map, red is used for unmapped reactions. A compound node appears in a color on a scale from yellow to green depicting the number of matched BIOCYC compounds (green for exact match). If a reaction node can still be mapped to one or more BIOCYC compounds even though in its neighborhood is at least one unmapped node, the reaction node appears blue. The saturation channel is used to indicate the ambiguity of the mapping. We make the distinction between reaction nodes with a completely mapped neighborhood vs. an incompletely mapped neighborhood, because with fewer node sets being taken into account when searching for a reaction match in the BIOCYC graph, the higher is the degree of freedom and the less reliable is the match.

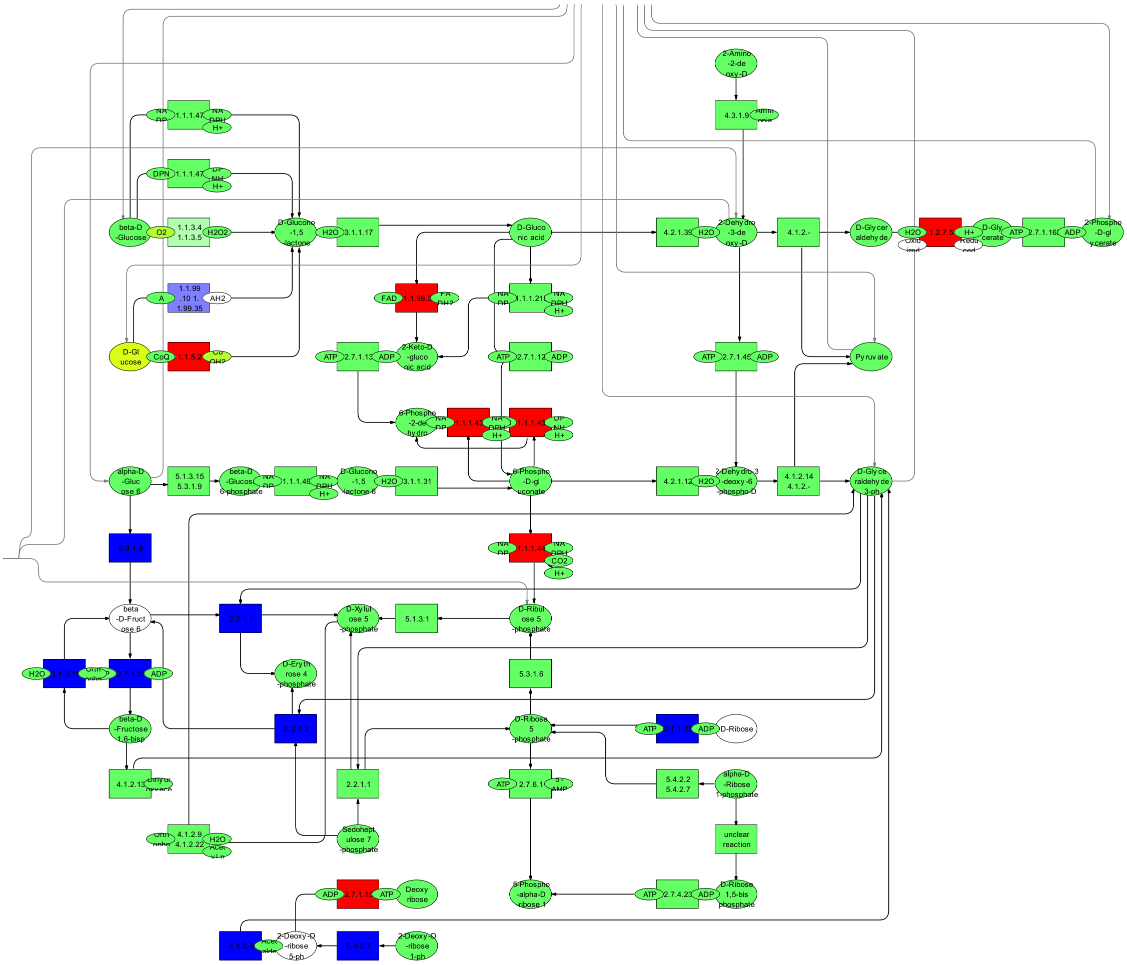


Figure 5.6: Match results on the detailed reaction network of the Pentose-Phosphate-Pathway. A color on the scale between yellow and green depicts the local match score, i.e., the inverse of the number of hits in the BioCyc graph. A white node color for compounds and red for reactions indicates, that the compound or reaction could not be found in the BioCyc graph. A reaction node drawn in blue indicates, that only a subset of adjacent compound nodes could be found in the BioCyc graph. In those cases, the reaction matches are much less reliable, but could very often verified using the EC number of the associated enzyme.

The user can finally verify the highlighted discrepancies in the two networks by displaying the context information in the BioCyc graph. For a selected reaction node, the mapped reaction(s) in BioCyc are displayed with the respective substrate and product compounds (Figure 5.7, r. h. s.). Vice versa, for a selected compound in KEGG, the corresponding match(es) with their adjacent reaction nodes are overlaid over the current graph (Figure 5.7, l. h. s.). The aforementioned node coloring scheme is applied on the BioCyc nodes as well. In addition to the selected node's neighborhood, the partial pathway ontology containing the displayed reaction nodes is presented.

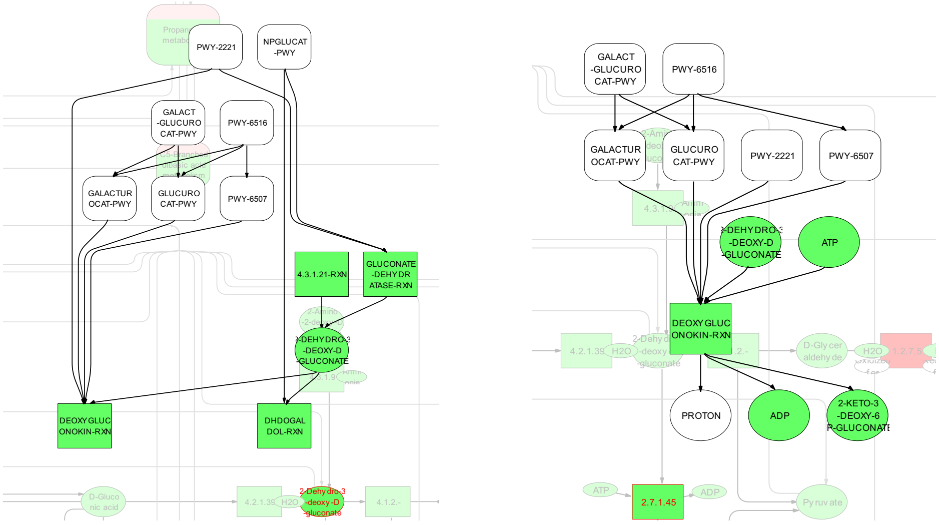


Figure 5.7: Details on demand: For the selected compound 2-Dehydro-3-deoxy-D-gluconate (l.h.s.) and the chemical reaction identified by the enzyme 2.7.1.45 (r.h.s.) of the Pentose-Phosphate-PW, the direct neighbors in the *Bio-Cyc* graph are displayed and the corresponding subset of the pathway ontology.

Displaying the KEGG template graph and the reaction graph context in BioCyc simultaneously has another advantage besides mental map preservation. This choice of design allows the user to edit the KEGG pathway graph to correct flaws in the network. The graph editing capability of the software tool allows manipulating the graph's topology as well as assigning attributes to graph elements.

5.2.5 Results

We constructed two different KEGG networks as template graph and ran the comparison on both graphs. The first network contains the complete set of available pathways (140). The second is a compilation of 15 pathways related to the carbohydrate metabolism. For experimental purposes, we ran two scenarios on that smaller network. The pathways provided by KEGG usually do not contain chemical compounds participating in a large number of reactions, e.g., H_2O, ATP, ADP, CO_2. Because of their high abundance, it is meaningful to exclude them from the visualization. However, these compounds greatly influence the quality of the reaction mapping, as these are obviously present in the BioCyc network. That fact becomes apparent when comparing Figures 5.6 and 5.8. In the first scenario, ubiquitous molecules were artificially added to the neighborhood of every reaction node according to the reaction's specific substrate and product sets. The second scenario neglects the presence of these compounds resulting in an improved visualization. In this way, the reaction's neighborhood constraint for a match is weakened, and therefore, leads to more ambiguous mappings. It is easy to conclude that many of the matches present in the simplified network are indeed mismatches as they do not occur in the scenario with stronger neighborhood conditions. Table 5.2 supports this observation.

This network contained 552 chemical compounds including ubiquitous molecules of which 455 could be mapped to at least one BioCyc compound. 471 of 616 reactions were identified in the BioCyc network. In the case of the simplified graph, 380 out of 449 compounds (84.6%) and 519 out of 616 reactions (84.3%) were mapped to BioCyc elements.

The observation of different mapping qualities depending on the in- or exclusion of ubiquitous metabolites can be verified in the detailed view as depicted in Figure 5.9. We could yet make another discovery on the reaction level. In some cases of ambiguous mappings, a perfect match would indeed be a mapping error! As the context graph in Figure 5.10 shows, the KEGG compound 'D-Glucose' is mapped to two Glucose nodes in BioCyc: 'D-Glucose' and 'GLC'. Referring to the annotation, the KEGG compound is more precisely *alpha-D-Glucose*, and GLC is *beta-D-Glucose*, a stereo-isomer of the former. Since no annotation is given

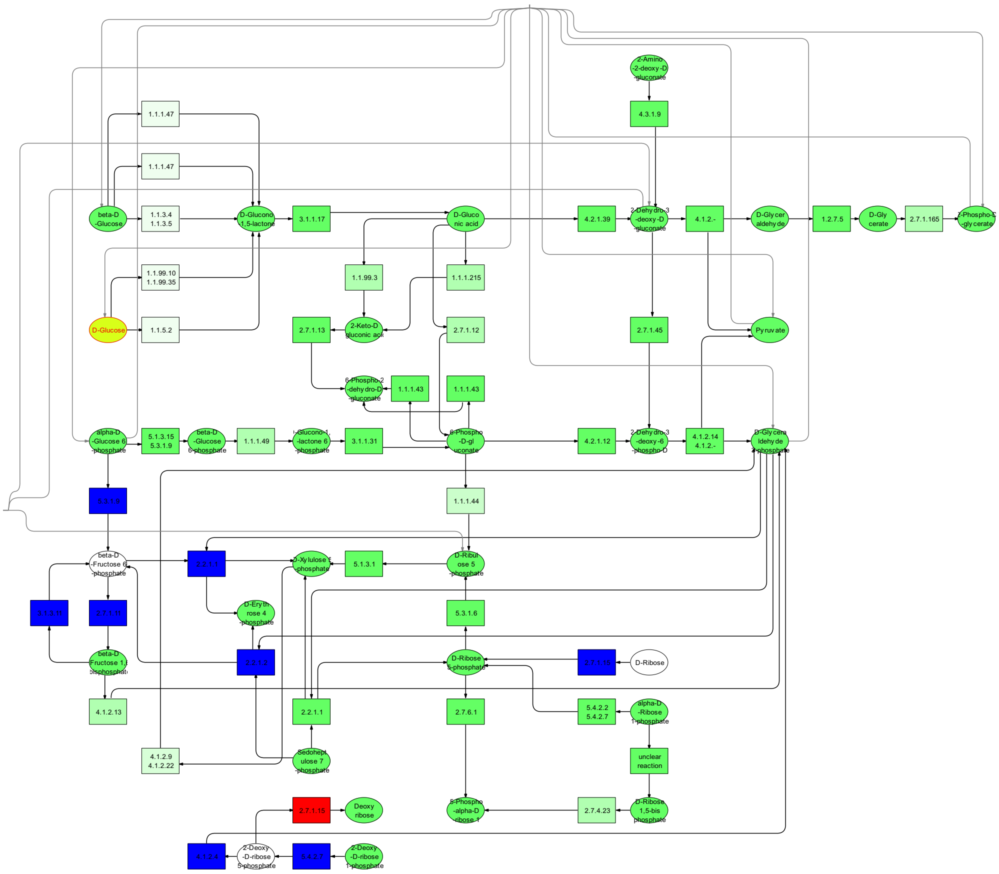


Figure 5.8: Match results on the detailed reaction network of the Pentose-Phosphate-Pathway. Unlike in Figure 5.6, ubiquitous molecules were omitted. This greatly improves the visualization. However, the reaction mapping is not as accurate as before. More reactions could be matched as suggested by the lower number of red-colored reaction nodes.

	Compounds			Reactions					total	
Pathway Name	#	f	s_p	#	f	s_p	f	s_p	s_p	$1-f$
Glycolysis / Gluconeogenesis	31	0.87	0.83	44	0.93	0.475	0.82	0.72	0.622	0.093
Citrate cycle (TCA cycle)	20	0.75	0.75	24	0.92	0.364	0.79	0.65	0.539	0.159
Pentose phosphate pathway	32	0.91	0.88	38	1.00	0.775	0.87	0.84	0.825	0.043
C5-Branched dibasic acid metabolism	32	0.75	0.75	32	0.69	0.589	0.66	0.60	0.669	0.281
Pentose and glucuronate interconv.	55	0.96	0.95	61	0.90	0.804	0.82	0.79	0.871	0.069
Propanoate metabolism	36	0.92	0.92	52	0.79	0.466	0.67	0.60	0.651	0.159
Fructose and mannose metabolism	48	0.88	0.87	63	0.81	0.718	0.76	0.74	0.782	0.162
Galactose metabolism	41	0.90	0.86	37	0.73	0.628	0.68	0.61	0.750	0.179
Ascorbate and aldarate metabolism	47	0.94	0.94	47	0.83	0.673	0.70	0.67	0.804	0.117
Starch and sucrose metabolism	50	0.72	0.70	69	0.70	0.543	0.65	0.57	0.607	0.294
Amino sugar and nucleotide sugar met.	87	0.84	0.81	95	0.74	0.672	0.69	0.67	0.739	0.214
Butanoate metabolism	40	0.85	0.85	53	0.87	0.632	0.74	0.70	0.726	0.140
Inositol phosphate metabolism	39	0.77	0.73	41	0.61	0.437	0.76	0.66	0.580	0.312
Pyruvate metabolism	32	0.88	0.88	65	0.94	0.419	0.77	0.70	0.570	0.083
Glyoxylate and dicarboxylate met.	50	0.98	0.96	66	0.97	0.637	0.88	0.84	0.776	0.026

Table 5.2: Summary of the mapping process. 84.6% of the compounds and 84.3% of the chemical reactions from the KEGG network could be mapped to nodes in the BioCyc network. f denotes the frequency of a match. For the reaction mapping, the first column contains match frequencies and scores in a graph without ubiquitous molecules (see Figure 5.8). This leaves a higher degree of freedom when checking the reaction's neighborhood condition. As a result, many ambiguous matches are found. In the latter reaction column we have the scenario in which ubiquitous molecules were present (compare with Figure 5.6). The last two columns summarise the overall mapping quality of the complete pathway without the ubiquitous metabolites. $1-f$ denotes the ratio of elements unable to map.

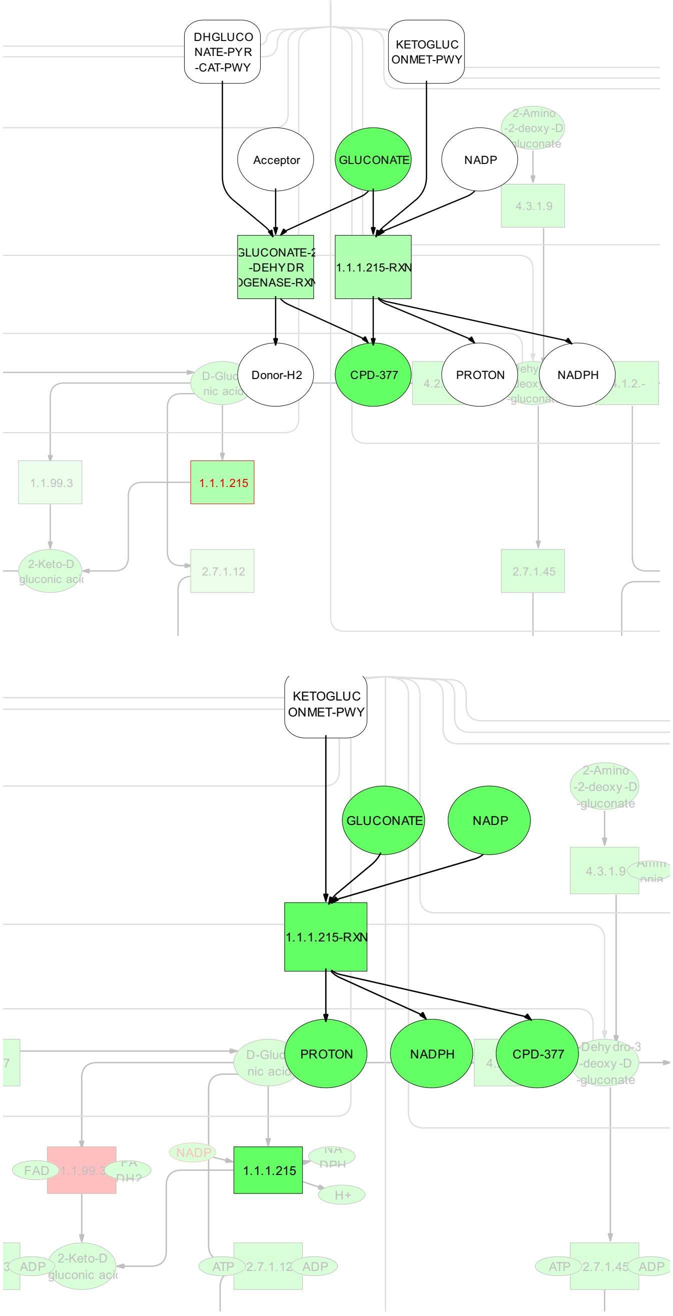


Figure 5.9: The number of compound nodes adjacent to reaction nodes influences the quality of the mapping. Top: Without the ubiquitous molecules (white elliptic nodes, not present in the KEGG network), the neighborhoods of the two reaction nodes are identical. Both reaction nodes will be mapped to the same reaction in KEGG.

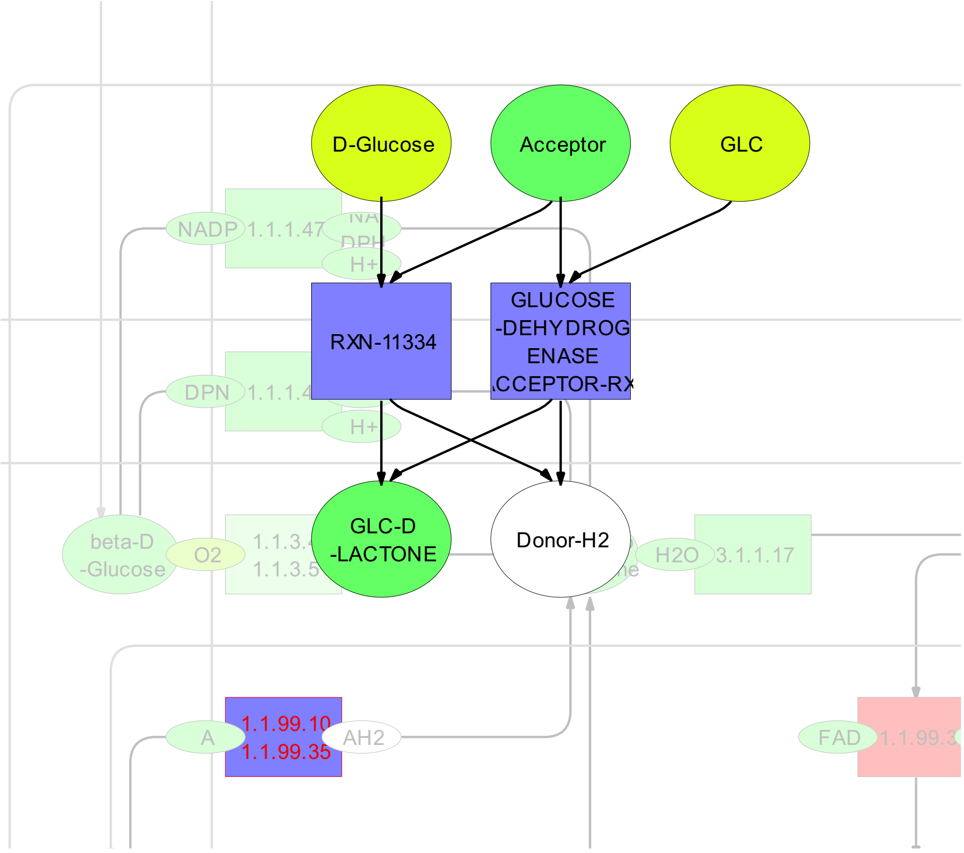


Figure 5.10: Example for an ambiguous match being closer to the ground truth than a unique match. Both types of *D-Glucose* are present in the network. The reaction node in KEGG represents a complex reaction consisting of several alternatives for metabolizing the different stereo-isomers of *D-Glucose*.

for the BioCyc node 'D-Glucose', we can only assume, that it represents indeed *alpha-D-Glucose* as specified in KEGG. In that case, the BioCyc pathway contains more – and more accurate – information, since both isomers are metabolized. In addition, the two reaction nodes mapped to the highlighted KEGG reaction are again a more precise description than provided by KEGG. The annotation for the KEGG reaction reveals two EC names, which indicate a complex reaction catalyzed by more than one enzyme. The two matches in BioCyc support this assumption as the reaction 'RXN-11334' corresponds to the enzyme number 1.1.99.35, and the reaction 'GLUCOSE-DEHYDROGENASE-ACCEPTOR-RXN' is identified with the EC number 1.1.99.10. Both EC numbers match the annotation of the respective reaction in KEGG.

For the complete metabolic network, 2872 out of 4817 compounds (60%) and 2169 out of 3129 reactions (69.3%) were identified in the BioCyc graph. The retrieval rates in the large network were considerably smaller than in the network representing the carbohydrate metabolism. This may have different reasons. Firstly, pathways related to the carbohydrate metabolism are well studied and understood. We can assume, that those pathways contain fewer errors. Secondly, among the more 'exotic' pathways in KEGG, there may be some reaction sets not present in the BioCyc collection at all.

5.2.6 Concluding remarks

In this work we have presented an extension to our graph visualization software capable of comparing metabolic networks from different bioinformatics resources. The network constructed from the KEGG Pathway database served as template graph and was used for navigating and exploring the data exploiting its 2-layer hierarchical structure. The reaction and compound nodes of this network were mapped to the metabolic network provided by the BioCyc database collection for the purpose of validation and verification of the KEGG data. The mapping quality was summarized using a simple, but yet meaningful score and presented to the user as an overview over the matched pathway graph. Discrepancies could be located and investigated in more detail taking advantage of the implemented focus&context technique described in chapter 3 for navigation and exploration.

Some differences in the datasets are plausible as explained in the results section, others are indeed incorrect entries or annotations. The proposed method helps to identify those problems, however, the evaluation of those discrepancies is the task of the user and certainly requires some background knowledge about the biochemical processes in question.

A very useful feature only shortly mentioned in this work is the graph editing capability of the software tool. By displaying portions of the BioCyc graph relevant to the selected elements in the KEGG graph on top of the current graph representation, the user can manually refine the metabolic network and save the changes.

There are, however, a few issues to be addressed. The mapping of compound nodes relies on the string comparison of the provided annotations and may miss matches due to syntactic errors in the synonym lists. Although the results suggest the matching process to be rather robust, manual refinement of the synonym lists may be necessary in some cases. On the other hand, identifying reaction nodes by their adjacency relations works very well even if some of the adjacent nodes could not be matched. In addition, perfect matches on reaction nodes with under-defined neighborhood can be exploited to match the missing compounds.

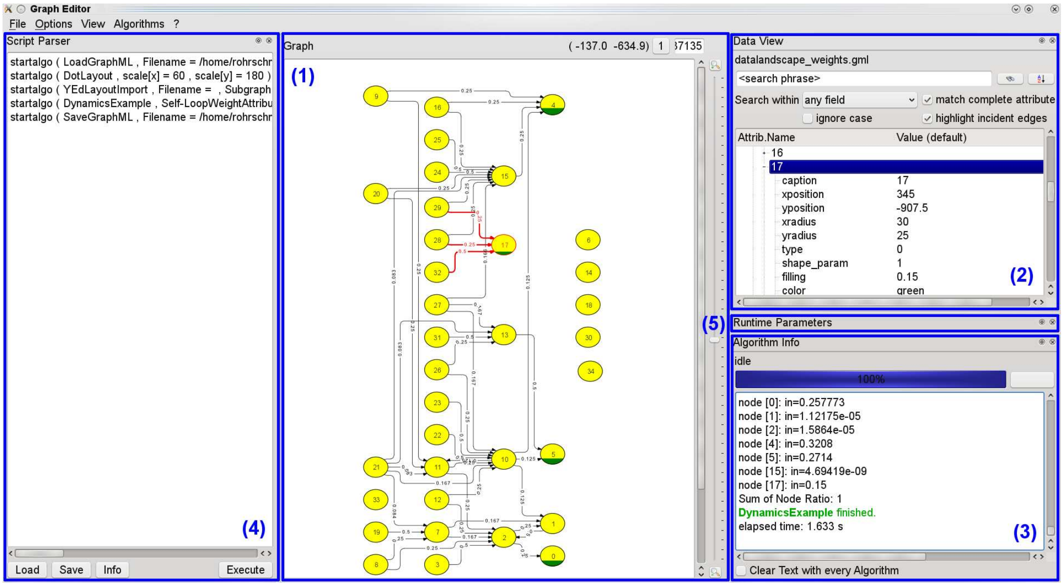
As stated in the previous section, the matching scores for 'exotic pathways' were very low suggesting that large parts of these pathways are missing in the BioCyc collection. This issue should be investigated further in close cooperation with bioinformatics experts and biologists.

6 Visualization Framework

All of the aforementioned routines were integrated into the software *GraphEdit*, an open-source graph visualization system based on the `Qt` (`qt.nokia.com`) framework. The software can be obtained from
`www.informatik.uni-leipzig.de/~hg/GraphEdit/`.
The system provides an editor component to construct graphs and manually refine the layout, as well as a GraphML parser for I/O. *GraphEdit* utilizes the graph library *libgraph*, a collection of template classes and functions developed by Christian Heine (`www.informatik.uni-leipzig.de/~hg/libgraph`).
GraphEdit contains a collection of algorithms for computing layouts and the appearance of graph elements, e.g., color, shape, size, as well as algorithms used for analysing specific graph properties. The software serves as development and test platform for new graph algorithms as it is easily extendable.
Although any kind of graphs (directed/undirected, multigraphs containing loops, hierarchical graphs, hypergraphs) are supported, the main emphasis was placed on the visualization and analysis of metabolic networks, which were modelled either as directed hierarchical hypergraphs, or directed bipartite graphs. There is a collection of very specific modules designed to address tasks on biochemical networks.

6.1 Graphical User Interface

The GUI of the visualization software provides five components implemented as dockable frames inside the main window, see Fig. 6.1:
1. The *Graph Scene* is the central component of the application's main window. The user has direct access to graph elements (nodes, edges, hyperedges, half-edges) to manipulate the graph's connectivity, hierarchical relations, and layout. Each graphical object rendered in the scene can individually be selected and appli-



```
startalgo ( LoadGraphML , Filename = /home/rohrschr
startalgo ( DotLayout , scale[x] = 60 , scale[y] = 180 )
startalgo ( YEdLayoutImport , Filename =  , Subgraph
startalgo ( DynamicsExample , Self-LoopWeightAttribu
startalgo ( SaveGraphML , Filename = /home/rohrschr
```

datalandscape_weights.gml

<search phrase>

Search within any field ☑ match complete attribute

☐ ignore case ☑ highlight incident edges

Attrib.Name	Value (default)
+ 16	
- 17	
caption	17
xposition	345
yposition	-907.5
xradius	30
yradius	25
type	0
shape_param	1
filling	0.15
color	green

idle

100%

```
node [0]: in=0.257773
node [1]: in=1.12175e-05
node [2]: in=1.5864e-05
node [4]: in=0.3208
node [5]: in=0.2714
node [15]: in=4.69419e-09
node [17]: in=0.15
Sum of Node Ratio: 1
DynamicsExample finished.
elapsed time: 1.633 s
```

☐ Clear Text with every Algorithm

Figure 6.1: Graphical User Interface of the visualization system. The central component of the application's main window is the *Graph Scene* (1) for direct access to the graph's connectivity and layout. The *Data View* (2) represents various properties of graph elements and nesting relations.

cable properties can be assigned via a context menu. The integrated graph editing capability allows the user to manually construct pathway graphs or to modify a given layout either generated by the algorithm or loaded from file. The component allows the user to change the view on the graph, e.g., by collapsing or expanding nodes, zooming, and panning.

2. The *Data View* component is a tree view displaying all associated – visual and non-visual – properties of graph elements and the hierarchical structure of the graph. The upper panel is used to perform a string-based search in the elements' attributes and to select element subsets.

3. 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.

4. The *Script Editor* allows the user to combine several algorithms to accomplish a complex task. Since all algorithms implemented in *GraphEdit* have access to the global instance of the data structure encapsulating the graph model, the result of one algorithm is the input of the following. All invoked algorithms are automatically

added to the editor as script commands keeping track of the call sequence.
5. The *Runtime Parameter Panel* is a placeholder for user-defined data entry masks. It is applicable to interactive algorithms described in the section below.

6.2 Algorithms in GraphEdit

All tasks manipulating the global instance of the graph data structure are implemented within classes derived from `GAlgorithm`. This ancestor class provides access to the graph scene and the algorithm info component as well as methods for registering the algorithm in *GraphEdit*. Upon invocation of an algorithm via the main window's menu, a user-defined dialog is shown for setting the initial parameters.
We distinguish between two kinds of algorithms, those requiring user intervention during execution and those that do not. Both types must implement the `doIt()` method, which is called upon invocation from the menu. In the latter case, the algorithm finishes after completion of this method. In the case of interactive algorithms, the `doIt()` method is only used for initialization and setting up the contents of the runtime parameter panel. For interacting with the graph in a task-specific way, we take advantage of `Qt`'s *Signal-Slot* implementation and connect event handlers to signals emitted by the graph scene informing about changes made in the graph by the user, for example, deletion of graph elements or changes in node positions or other associated attributes. Data entered into the mask held by the runtime parameter panel give the user the opportunity to control the progress of the algorithm and to influence the visualization of the results. Examples of interactive algorithms are described in chapters 4 and 5, whereas the layout computation described in chapter 3 is implemented as a non-interactive algorithm.

7 Conclusion

The goal of this thesis was to apply information visualization techniques combined with graph layout algorithms to networks representing metabolic network data. The systematic classification of chemical reactions into subsets, called metabolic pathways, is a standard procedure among biochemists. We took advantage of this classification to construct a metabolic network reflecting this hierarchical structure and used the hierarchy to make the complete network accessible in an interactive exploration process. One of the challenges was to integrate different scales – or levels of detail – into the model starting with the pathway level highlighting functional units within a biological cell's metabolism and biomass fluxes between them. The next level in this model is the reaction network representing the transformation of molecules by chemical reactions. Finally, each of the molecules itself can be considered as a graph representing the molecular structure. This hierarchical graph is augmented with domain-specific annotations, e.g., textual information as molecule or enzyme name, classification of enzymes, concentration values of molecules.

The first part of the thesis demonstrated, how this hierarchical system can be integrated into a visualization framework capturing all three levels of detail including the representation of the given annotations characterizing network elements in more detail. A suitable algorithm was presented to layout the pathway and reaction layers of the network and a technique derived from the *Table Lens Metaphor* was used to interactively explore the metabolic network data. The method was successfully applied to the complete set of chemical reactions taking place in a generalized eukaryotic cell.

While this static network represents the current *"as is"* state, little is known, how these networks may have developed. In the second part of the thesis, we examined metabolic network data from a simulation capturing evolutionary processes on a cell's metabolism. In these networks, chemical reactions occur, change, or disappear changing the metabolic network in terms of connectivity as well as with respect to the attribute values associated with the graph elements. The presented visual analysis process shed light on the formation of metabolic pathways and provided a deeper understanding of evolutionary mechanisms in early metabolism.

The third part of the project covers the comparison of metabolic networks. Here we have chosen two scenarios being of interest to the biological community: First, how does the metabolism of one selected species differs from another species, and second, what discrepancies are present in metabolic network data obtained from different resources. The latter addresses the problem of data quality and consistency among different data bases, which has not been investigated in that detail before.

In the process of the work, a graph visualization application was developed, which was used to implement and demonstrate the aforementioned methods and algorithms.

8 Appendix

The following table lists all metabolic pathways obtained from KEGG giving an overview on the number of organisms realizing a particular pathway, and the number of different pathway realizations. The first number in the last column counts different realizations directly taken from the KEGG Organismdata base. The main source of differences among the realizations is the presence of isolated nodes. Although carrying meaning for the presentation in KEGG, we consider them as artifacts in the network. After removal of these nodes, the number of realizations is greatly reduced as shown in the last column.

Pathway name (cont.)	ID	#Spp.	Realizations	
Glycolysis / Gluconeogenesis	00010	1366	36	1
Citrate cycle (TCA cycle)	00020	1339	61	1
Pentose phosphate pathway	00030	1357	42	2
Pentose and glucuronate interconversions	00040	1261	47	2
Fructose and mannose metabolism	00051	1353	34	2
Galactose metabolism	00052	1272	36	2
Ascorbate and aldarate metabolism	00053	892	35	2
Fatty acid biosynthesis	00061	1285	25	1
Fatty acid elongation	00062	121	1	1
Fatty acid metabolism	00071	1178	30	2
Synthesis and degradation of ketone bodies	00072	719	18	1
Steroid biosynthesis	00100	263	5	2
Primary bile acid biosynthesis	00120	18	2	2
Secondary bile acid biosynthesis	00121	218	11	2
Ubiquinone and other terpenoid-quinone biosynthesis	00130	1213	33	2
Steroid hormone biosynthesis	00140	18	2	2
Purine metabolism	00230	1370	38	2
Puromycin biosynthesis	00231	1	1	1
Caffeine metabolism	00232	85	5	2
Pyrimidine metabolism	00240	1370	32	2
Alanine, aspartate and glutamate metabolism	00250	1336	80	2
Tetracycline biosynthesis	00253	26	2	2

Pathway name (cont.)	ID	#Spp.	Realizations	
Glycine, serine and threonine metabolism	00260	1352	77	2
Cysteine and methionine metabolism	00270	1363	54	2
Valine, leucine and isoleucine degradation	00280	1357	36	2
Geraniol degradation	00281	642	14	2
Valine, leucine and isoleucine biosynthesis	00290	1370	25	2
Lysine biosynthesis	00300	1309	34	1
Lysine degradation	00310	1134	25	2
Penicillin and cephalosporin biosynthesis	00311	273	9	2
beta-Lactam resistance	00312	31	13	1
Arginine and proline metabolism	00330	1334	60	2
Clavulanic acid biosynthesis	00331	3	2	2
Histidine metabolism	00340	1188	25	2
Tyrosine metabolism	00350	1265	50	2
Trichloro-2,2-bis(4-chlorophenyl)ethane degradation	00351	8	2	2
Phenylalanine metabolism	00360	1147	55	2
Chlorocyclohexane and chlorobenzene degradation	00361	532	19	2
Benzoate degradation	00362	905	40	2
Bisphenol degradation	00363	272	14	2
Fluorobenzoate degradation	00364	465	18	2
Tryptophan metabolism	00380	1261	34	2
Phenylalanine, tyrosine and tryptophan biosynthesis	00400	1312	74	1
Novobiocin biosynthesis	00401	1018	21	2
Benzoxazinoid biosynthesis	00402	2	1	1
beta-Alanine metabolism	00410	1117	28	2
Taurine and hypotaurine metabolism	00430	1074	25	2
Phosphonate and phosphinate metabolism	00440	433	11	2
Selenocompound metabolism	00450	1357	32	2
Cyanoamino acid metabolism	00460	1145	48	2
D-Glutamine and D-glutamate metabolism	00471	1132	30	2
D-Arginine and D-ornithine metabolism	00472	351	5	2
D-Alanine metabolism	00473	1034	31	2
Glutathione metabolism	00480	1345	34	2
Starch and sucrose metabolism	00500	1325	41	2
N-Glycan biosynthesis	00510	139	11	1
Other glycan degradation	00511	636	266	10
Mucin type O-glycan biosynthesis	00512	49	2	1
Various types of N-glycan biosynthesis	00513	40	28	2
Other types of O-glycan biosynthesis	00514	99	30	1
Amino sugar and nucleotide sugar metabolism	00520	1344	74	2
Streptomycin biosynthesis	00521	1098	25	2
Biosynthesis of 12-, 14- and 16-membered macrolides	00522	3	2	2
Polyketide sugar unit biosynthesis	00523	882	18	2
Butirosin and neomycin biosynthesis	00524	22	2	2
Glycosaminoglycan degradation	00531	102	30	2
Glycosaminoglycan biosynthesis - chondroitin sulfate	00532	49	34	1
Glycosaminoglycan biosynthesis - keratan sulfate	00533	39	29	7
Glycosaminoglycan biosynthesis - heparan sulfate	00534	49	44	1

Pathway name (cont.)	ID	#Spp.	Realizations	
Lipopolysaccharide biosynthesis	00540	721	79	1
Peptidoglycan biosynthesis	00550	1093	234	2
Glycerolipid metabolism	00561	1269	29	2
Inositol phosphate metabolism	00562	1202	28	2
Glycosylphosphatidylinositol(GPI)-anchor biosynthesis	00563	138	40	1
Glycerophospholipid metabolism	00564	1354	37	2
Ether lipid metabolism	00565	190	8	2
Arachidonic acid metabolism	00590	768	14	2
Linoleic acid metabolism	00591	79	3	2
alpha-Linolenic acid metabolism	00592	664	8	2
Sphingolipid metabolism	00600	481	14	2
Glycosphingolipid biosynthesis - lacto and neolacto series	00601	51	4	2
Glycosphingolipid biosynthesis - globo series	00603	75	6	2
Glycosphingolipid biosynthesis - ganglio series	00604	73	5	2
Pyruvate metabolism	00620	1368	66	2
Dioxin degradation	00621	414	19	2
Xylene degradation	00622	297	14	3
Toluene degradation	00623	526	19	2
Polycyclic aromatic hydrocarbon degradation	00624	725	13	2
Chloroalkane and chloroalkene degradation	00625	1000	27	2
Naphthalene degradation	00626	930	20	2
Aminobenzoate degradation	00627	849	38	2
Glyoxylate and dicarboxylate metabolism	00630	1346	49	1
Nitrotoluene degradation	00633	602	15	2
Propanoate metabolism	00640	1355	42	2
Ethylbenzene degradation	00642	240	15	4
Styrene degradation	00643	520	14	1
Butanoate metabolism	00650	1326	45	2
C5-Branched dibasic acid metabolism	00660	1035	34	2
One carbon pool by folate	00670	1357	17	2
Methane metabolism	00680	1275	62	2
Carbon fixation in photosynthetic organisms	00710	119	5	1
Carbon fixation pathways in prokaryotes	00720	12	2	1
Thiamine metabolism	00730	1317	36	2
Riboflavin metabolism	00740	1314	31	2
Vitamin B6 metabolism	00750	1281	32	2
Nicotinate and nicotinamide metabolism	00760	1348	36	2
Pantothenate and CoA biosynthesis	00770	1345	40	2
Biotin metabolism	00780	1265	9	2
Lipoic acid metabolism	00785	1206	15	1
Folate biosynthesis	00790	1304	36	2
Atrazine degradation	00791	216	4	2
Retinol metabolism	00830	49	13	2
Porphyrin and chlorophyll metabolism	00860	1300	19	2
Terpenoid backbone biosynthesis	00900	1331	44	2
Indole alkaloid biosynthesis	00901	2	2	2
Monoterpenoid biosynthesis	00902	2	2	2

Pathway name (cont.)	ID	#Spp.	Realizations	
Limonene and pinene degradation	00903	829	10	2
Diterpenoid biosynthesis	00904	9	4	4
Brassinosteroid biosynthesis	00905	8	2	2
Carotenoid biosynthesis	00906	271	13	3
Zeatin biosynthesis	00908	9	3	3
Sesquiterpenoid biosynthesis	00909	4	2	2
Nitrogen metabolism	00910	1312	36	2
Sulfur metabolism	00920	1221	29	2
Caprolactam degradation	00930	490	9	2
Phenylpropanoid biosynthesis	00940	9	3	3
Flavonoid biosynthesis	00941	9	3	3
Anthocyanin biosynthesis	00942	3	2	2
Isoflavonoid biosynthesis	00943	1	1	1
Flavone and flavonol biosynthesis	00944	8	2	2
Stilbenoid, diarylheptanoid and gingerol biosynthesis	00945	9	3	3
Isoquinoline alkaloid biosynthesis	00950	12	2	2
Tropane, piperidine and pyridine alkaloid biosynthesis	00960	14	2	2
Betalain biosynthesis	00965	2	2	2
Glucosinolate biosynthesis	00966	2	2	2
Aminoacyl-tRNA biosynthesis	00970	1372	791	2
Metabolism of xenobiotics by cytochrome P450	00980	49	3	2
Insect hormone biosynthesis	00981	21	2	2
Drug metabolism - cytochrome P450	00982	49	6	2
Drug metabolism - other enzymes	00983	49	8	2

References

[1] KEGG Pathway Online Documention.
http://www.genome.jp/kegg/document/help_pathway.html.

[2] MDL MOL format.
http://www2.chemie.uni-erlangen.de/services/vsc/db_vsc/remarks/mol_subst.html.

[3] M. Albrecht, A. Estrella-Balderrama, M. Geyer, C. Gutwenger, K. Klein, O. Kohlbacher, and M. Schulz. 08191 working group summary – visually comparing a set of graphs. In *Graph Drawing with Applications to Bioinformatics and Social Sciences*, number 08191 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[4] M. Albrecht, A. Kerren, K. Klein, O. Kohlbacher, P. Mutzel, W. Paul, F. Schreiber, and M. Wybrow. A Graph-drawing Perspective to Some Open Problems in Molecular Biology. Technical Report TR08-01-003, Lehrstuhl XI für Algorithm Engineering, Fakultät für Informatik, Technische Universität Dortmund, Germany, 2008.

[5] M. Albrecht, A. Kerren, K. Klein, O. Kohlbacher, P. Mutzel, W. Paul, F. Schreiber, and M. Wybrow. On open problems in biological network visualization. In *Proc. International Symposium on Graph Drawing (GD '09)*, volume 5849 of *LNCS*, pages 256–267. Springer, 2010.

[6] M. Y. Becker and I. Rojas. A graph layout algorithm for drawing metabolic pathways. *Bioinformatics*, 17(5):461–467, 2001.

[7] S. Bender-DeMoll and D. McFarland. The Art and Science of Dynamic Network Visualization. *JoSS: Journal of Social Structure*, 7, 2005.

[8] U. Brandes, T. Dwyer, and F. Schreiber. Visualizing related metabolic pathways in two and a half dimensions. In G. Liotta, editor, *Proc. International Symposium on Graph Drawing (GD'03)*, volume 2912 of *LNCS*, pages 111–122, 2003.

[9] U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, and M. Scott Marshall. Graphml progress report: Structural layer proposal. In *Proceedings of the 9th International Symposium on Graph Drawing (GD 2001)*, volume 2265 of *LNCS*, pages 501–512. Springer, 2002.

[10] J. Branke. Dynamic graph drawing. In *Drawing graphs: methods and models*, pages 228–246, London, UK, 2001. Springer-Verlag.

[11] G. Caetano-Anollés, L. S. Yafremava, H. Gee, D. Caetano-Anollés, Hee S. Kim, and Jay E. Mittenthal. The origin and evolution of modern metabolism. *The International Journal of Biochemistry & Cell Biology*, 41(2):285–297, February 2009.

[12] R. Caspi, T. Altman, J.M. Dale, K. Dreher, C.A. Fulcher, F. Gilham, P. Kaipa, A.S. Karthikeyan, A. Kothari, M. Krummenacker, M. Latendresse, L.A. Mueller, S. Paley, L. Popescu, A. Pujar, A. Shearer, P. Zhang, and P.D. Karp. Metacyc: The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research*, 38:D473–D479, 2010.

[13] R. Caspi, H. Foerster, C. A. Fulcher, R. Hopkinson, J. Ingraham, P. Kaipa, M. Krummenacker, S. Paley, J. Pick, S. Y. Rhee, C. Tissier, P. Zhang, and P. D. Karp. Metacyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Research*, 34(Database issue):D511–D516, 2006.

[14] F Cordon. *Tratado evolucionista de biologia.* Aguilar Ediciones, Madrid, Spain, 1990.

[15] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice Hall, New Jersey, 1999.

[16] S. Diehl and C. Görg. Graphs, they are changing - dynamic graph drawing for a sequence of graphs. In *Proc. 10th Int. Symp. Graph Drawing (GD 2002), number 2528 in Lecture Notes in Computer Science, LNCS*, pages 23–31. Springer-Verlag, 2002.

[17] S. Diehl, C. Görg, and A. Kerren. Preserving the mental map using foresighted layout. In *Proceedings of Joint Eurographics-IEEE TVCG Symposium on Visualization, VisSym 2001*, pages 175–184. Springer, 2001.

[18] M.A. Doyle, J.I. MacRae, D.P. De Souza, E.C. Saunders, and M.J. McConville V.A. Likić. LeishCyc: a biochemical pathways database for *Leishmania major*. *BMC Syst Biol.*, 3:57, 2009.

[19] C. Erten, P. J. Harding, S. G. Kobourov, Ke. Wampler, and G. V. Yee. Graphael: Graph animations with evolving layouts. In G. Liotta, editor, *Graph Drawing*, volume 2912 of *Lecture Notes in Computer Science*, pages 98–110. Springer, 2003.

[20] C. Erten, S. G. Kobourov, and C. Pitta. Morphing planar graphs. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 451–452, New York, NY, USA, 2004. ACM.

[21] J.-L. Faulon and A. G. Sault. Stochastic generator of chemical structure. 3. Reaction network generation. *J. Chem. Inf. Comp. Sci.*, 41(4):894–908, 2001.

[22] V.C. Forst, C. Flamm, I.L. Hofacker, and P.F. Stadler. Algebraic comparison of metabolic networks, phylogenetic inference, and metabolic innovation. *BMC Bioinformatics*, 7:67 [epub], 2006.

[23] Y. Frishman and A. Tal. Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):727–740, 2008.

[24] J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5(175), 2004.

[25] E. R. Gansner, E. Koutsofios, S. C. North, and Kiem-Phong Vo. A technique for drawing directed graphs. *IEEE Trans. Software Eng.*, 19(3):214–230, 1993.

[26] A. Gevorgyan, M.G. Poolman, and D.A. Fell. Detection of stoichiometric inconsistencies in biomolecular models. *Bioinformatics*, 24:2245–2251, 2008.

[27] A. Goesmann, M. Haubrock, F. Meyer, J. Kalinowski, and R. Giegerich. Pathfinder: reconstruction and dynamic visualization of metabolic pathways. *Bioinformatics*, 18(1):124–129, 2002.

[28] M. L. Green and P. D. Karp. The outcomes of pathway database computations depend on pathway ontology. *Nucleic Acids Research*, 34(13):3687–3697, 2006.

[29] M.L. Green and P.D. Karp. Genome annotation errors in pathway databases due to semantic ambiguity in partial EC numbers. *Nucleic Acids Res.*, 33:4035–4039, 2005.

[30] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[31] R.A. Haw, D. Croft, C.K. Yung, N. Ndegwa, P. D'Eustachio, H. Hermjakob, and L.D. Stein. The Reactome BioMart. *Database*, 2011, 2011.

[32] Joshua Wing Kei Ho, T. Manwaring, Seok-Hee Hong, U. Roehm, David Cho Yau Fung, Kai Xu, T. Kraska, and D. Hart. Pathbank: Web-based querying and visualization of an integrated biological pathway database. In *CGIV '06: Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, pages 84–89, Washington, DC, USA, 2006. IEEE Computer Society.

[33] N. H. Horowitz. On the evolution of biochemical syntheses. *Proc Natl Acad Sci USA*, 31:153–157, 1945.

[34] Z. Hu, J. Mellor, J. Wu, and C. DeLisi. VisANT: an online visualization and analysis tool for biological interaction data. *BMCBIOINF*, 5(1):e17, 2004.

[35] R. A. Jensen. Enzyme recruitment in evolution of new function. *Annu Rev Microbiol*, 30:409–425, 1976.

[36] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1989.

[37] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, 36(Database issue):D480–D484, 2008.

[38] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[39] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(Database-Issue):354–357, 2006.

[40] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahren, S. Tsoka, N. Darzentas, V. Kunin, and N. Lopez-Bigas. Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Research*, 33(19):6083–6089, 2005.

[41] P. D. Karp, S. M. Paley, and P. Romero. The pathway tools software. *Bioinformatics*, 18:225–232, 2002.

[42] KEGG. Kyoto Encyclopedia of Genes and Genomes.
http://www.kegg.jp/kegg/.

[43] A. Kerren. 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.

[44] KGML. KEGG Markup Language.
http://www.genome.jp/kegg/docs/xml/.

[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[46] C. Klukas and F. Schreiber. Dynamic exploration and editing of KEGG pathway diagrams. *Bioinformatics*, 23(3):344–350, 2007.

[47] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):805–812, 2006.

[48] A. Lubiw, M. Petrick, and M. Spriggs. Morphing orthogonal planar graph drawings. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 222–230, New York, NY, USA, 2006. ACM.

[49] M. Lungu and K. Xu. Biomedical Information Visualization. In *Human-Centered Visualization Environments*, pages 311–342, 2006.

[50] G. Michal. *Biochemical Pathways: Biochemie-Atlas.* Spektrum Akad. Verlag, Heidelberg, 1999.

[51] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6(2):183–210, 1995.

[52] J. Moody, D. McFarland, and S. Bender-DeMoll. Dynamic network visualization. *American Journal of Sociology*, 110(4), 2005.

[53] L. F. Moreno. Understanding Fischer Projection and Angular Line Representation Conversion. *J. Chem. Educ.*, (1):175–176, 2012.

[54] D. E. Nicholson. *Metabolic Pathways Map (Poster).* Sigma Chemical Co., St. Louis, 1997.

[55] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 27(1):29–34, 1999.

[56] M. Oldiges, S. Noack, A. Wahl, E. Qeli, B. Freisleben, and W. Wiechert. From enzyme kinetics to metabolic network modeling - visualization tool for enhanced kinetic analysis of biochemical network models. *Eng. Life Sci.*, 6(2), 2006.

[57] M.A. Ott and G. Vriend. Correcting ligands, metabolites, and pathways. *BMC Bioinformatics*, 7:517, 2006.

[58] B. O. Palsson. *Systems Biology: Properties of Reconstructed Networks.* Cambridge University Press, New York, NY, USA, 2006.

[59] A.R. Pico, T. Kelder, M.P. van Iersel, K. Hanspers, B.R. Conklin, and C. Evelo. WikiPathways: Pathway Editing for the People. *PLoS Biology*, 6(7):1403–1407, 2008.

[60] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus&context visualization for tabular information. In *CHI '94: Conference companion on Human factors in computing systems*, page 222. ACM, 1994.

[61] J. C. Roberts. Exploratory visualization with multiple linked views. In *Exploring Geovisualization.* Amsterdam: Elseviers, December 2004.

[62] M. Rohrschneider, C. Heine, A. Reichenbach, A. Kerren, and G. Scheuermann. A Novel Grid-based Visualization Approach for Metabolic Networks with Advanced Focus & Context View. In *Proceedings of the 17th International Symposium on Graph Drawing (GD 2009)*, volume 5849 of *LNCS*, pages 268–279. Springer, 2010.

[63] I. Rojdestvenski. VRML metabolic network visualizer. *Computers in Biology and Medicine*, 33(2):169–182, 2003.

[64] P. Saraiya, Ch. North, and K. Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization*, 4(3):191–205, 2005.

[65] A.M. Schnoes, S.D. Brown, I. Dodevski, and P.C. Babbitt. Annotation error in public databases: Misannotation of molecular function in enzyme superfamilies. *PLoS Comput Biol*, 5:e1000605, 2009.

[66] F. Schreiber. High Quality Visualization of Biochemical Pathways in BioPath. *In Silico Biology*, 2(2):59–73, 2002.

[67] F. Schreiber. Visual comparison of metabolic pathways. *J. Vis. Lang. Comput.*, 14(4):327–340, 2003.

[68] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003.

[69] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[70] R. Steuer, T. Gross, J. Selbig, and B. Blasius. Structural kinetic modeling of metabolic networks. *Proc. Natl. Acad. Sci.*, 103(32):11868–11873, 2006.

[71] M. Streit, M.Kalkusch, K. Kashofer, and D. Schmalstieg. Navigation and exploration of interconnected pathways. *Eurographics / IEEE-VGTC Symposium on Visualization*, 27(3), 2008.

[72] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical systems. *IEEE Trans. Systems, Man, and Cybernetics*, 11:109–125, 1981.

[73] A. Ullrich and C. Flamm. Functional evolution of ribozyme-catalyzed metabolisms in a graph-based toy-universe. In S. Istrail, editor, *Proceedings of the 6th International Conference on Computational Methodes in Systems Biology (CSMB)*, volume 5307 of *Lect. Notes Bioinf.*, pages 28–43, 2008.

[74] A. Ullrich and C. Flamm. A sequence-to-function map for ribozyme-catalyzed metabolisms. In *ECAL*, volume 5777 of *Lect. Notes Comp. Sci.*, 2009.

[75] A. Ullrich, M. Rohrschneider, G. Scheuermann, P. F. Stadler, and C. Flamm. In silico evolution of early metabolism. *Artificial Life*, 17(2):87–108, 2011.

[76] D. Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28:31–36, 1988.

[77] H. Wiener. Structural determination of paraffin boiling points. *Journal of the American Chemical Society*, 69(1):17–20, 1947.

[78] Kun Yang, Wenzhe Ma, Huanhuan Liang, Qi Ouyang, Chao Tang, and Luhua Lai. Dynamic simulations on the arachidonic acid metabolic network. *PLoS Comput Biol*, 3(3):e55, 03 2007.

[79] D. Zhu and Z.S. Qin. Structural comparison of metabolic networks in selected single cell organisms. *BMCBIOINF*, 6(8), 2005.

Curriculum Vitae

Persönliche Daten

Name: Rohrschneider
Vorname: Markus
Geburtsdatum: 14.08.1977
Geburtsort: Leipzig
Nationalität: deutsch
Eltern: Dr. med. Gottfried Rohrschneider
Christine Rohrschneider, geb. Grunert

Schulbildung

1984	Einschulung in die 54. Oberschule “Victor Jara”
1992 - 1994	Besuch des Kant-Gymnasiums mit naturwissenschaftlich / musischem Profil
1994 - 1995	High School Aufenthalt in Pittsburgh, PA
1995 - 1997	Besuch des Kant-Gymnasiums
1997	Abitur

Zivildienst

1997 - 1998	Unfallchirurgie des Universitätsklinikums Leipzig

Studium

10/1998	Beginn des Studiums der Humanmedizin an der Universität Leipzig
10/2000	Beginn des Studiums der Informatik Nebenfach Biowissenschaften (Universität Leipzig)
07/2003	Vordiplom
01/2007	Abschluss des Informatikstudiums (Diplom)

Beruflicher Werdegang

02/2007	Wissenschaftliche Hilfskraft am Institut für Informatik, Universität Leipzig
05/2007 - 04/2012	Wissenschaftlicher Mitarbeiter (Doktorand) am Institut für Informatik, Universität Leipzig
06/2011	Gründung der Effigos AG in Leipzig
08/2012 - 08/2013	Softwareentwickler bei der USK Karl Utz Sondermaschinen GmbH
seit 06/2011	Wissenschaftlicher Vorstand (CRO) der Effigos AG

List of publications

- Markus Rohrschneider, Gerik Scheuermann, Stefan Hoehme, Dirk Drasdo. Shape Characterization of Extracted and Simulated Tumor Samples using Topological and Geometric Measures. In: *IEEE Engineering in Medicine and Biology Conference 2007 Proceedings*, pages 6271-6277 (2007).
- Markus Rohrschneider, Christian Heine, André Reichenbach, Andreas Kerren, Gerik Scheuermann. A Novel Grid-based Visualization Approach for Metabolic Networks with Advanced Focus & Context View. In: *Proceedings of the 17th International Symposium on Graph Drawing (GD 2009)*, LNCS 5849: 268-279 (2010).
- Markus Rohrschneider, Alexander Ullrich, Andreas Kerren, Peter F. Stadler, and Gerik Scheuermann. Visual Network Analysis of Dynamic Metabolic Pathways. In: *Proceedings of the 6th International Symposium on Visual Computing (ISVC 2010)*, LNCS 6453: 316-327 (2010).
- Christoph Flamm, Alexander Ullrich, Heinz Ekker, Martin Mann, Daniel Högerl, Markus Rohrschneider, Sebastian Sauer, Gerik Scheuermann, Konstantin Klemm, Ivo L. Hofacker, Peter F. Stadler. Evolution of Metabolic Networks: A Computational Framework. In: *Journal of Systems Chemistry*, 1: 4 (2010).
- Alexander Ullrich, Markus Rohrschneider, Gerik Scheuermann, Peter F. Stadler, Christoph Flamm. In silico evolution of early metabolism. In: *Artificial Life*, 17 (2): 87-108 (2011).
- Markus Rohrschneider, Peter F. Stadler, Gerik Scheuermann. A Visual Cross-Database Comparison of Metabolic Networks. In: *Proceedings of the 8th International Symposium on Visual Computing (ISVC 2012)*, LNCS 7432, 678-687 (2012).