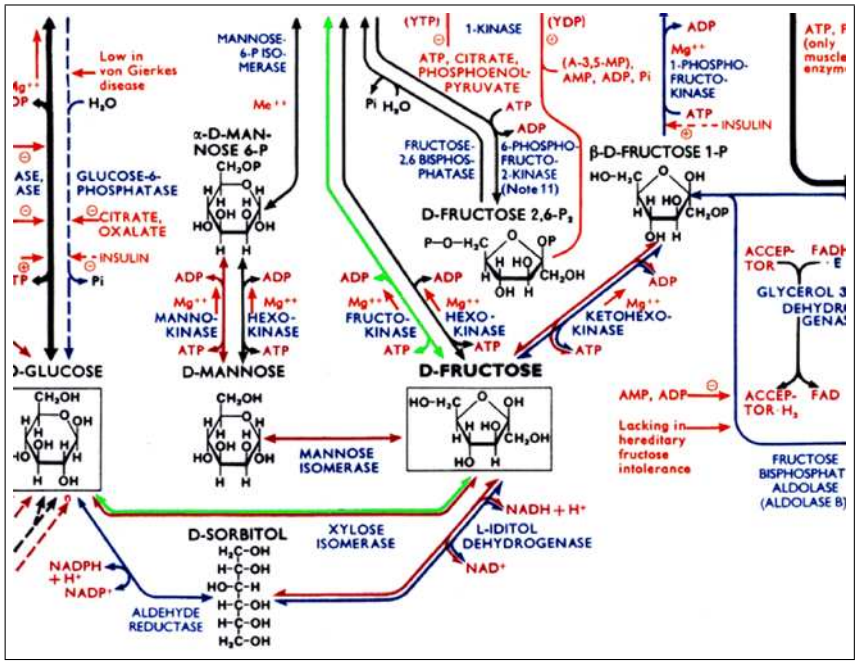
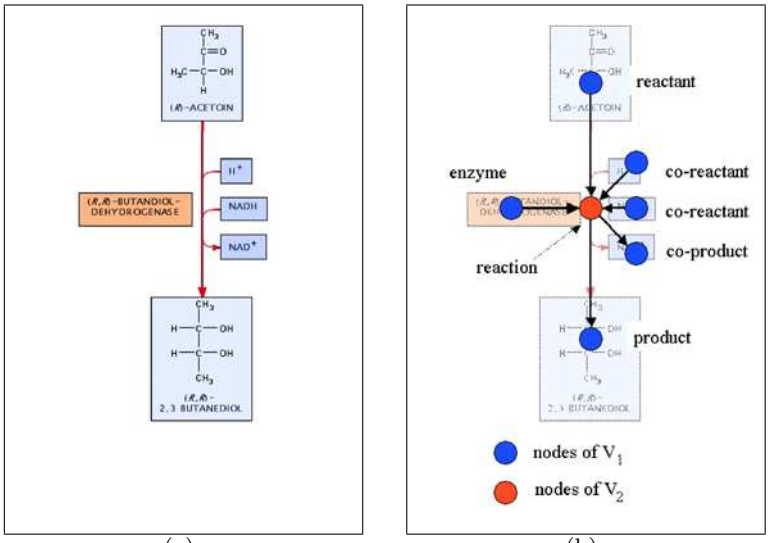


Fig. 1. A cutting of the *Boehringer Biochemical Pathways* poster [16] showing the complexity of the network

there is more knowledge than this aspect and different views are needed. In the extreme, one needs an overview diagram with a detailed view of some parts such as an overview diagram of all catabolic pathways with the citrate cycle (TCA cycle) in a detailed view. Using static visualization the user is not able to interactively change the depth of information in the diagram, and cannot browse through pathways from abstract overview diagrams to detailed views.

Formally, a biochemical pathway is a directed hyper-graph consisting of vertices and hyper-edges. Let $\wp(V)$ denote the power set of V. A directed hyper-graph $G = (V, E)$ consists of a finite set $V = V(G)$ of vertices and a finite set $E \subseteq \wp(V) \times \wp(V)$ of directed hyper-edges. Compared with edges in directed graphs defined in subsection 2.2 of the Technical Foundations Chapter, hyper-edges may connect more than two vertices. The vertices represent the substances, co-substances, and enzymes within a pathway. A *substance* can be a reactant or product; a *co-substance* can be a co-reactant or a co-product. A *compound* is the generic term for all elements of a reaction: substances, co-substances, and enzymes. Each reaction is represented by a directed hyper-edge, which connects all compounds of the reaction, see Fig. 2(a). More common is the modeling of such relations by directed bipar-

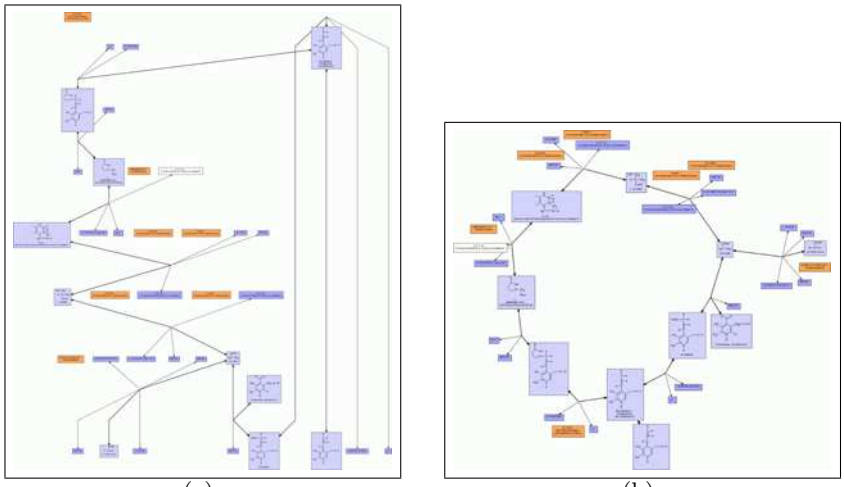


(a) (b)

Fig. 2. The diagram **(a)** shows a biochemical reaction. The connections between enzymes and other compounds are usually displayed by placing an enzyme close to the corresponding hyper-edge. Image **(b)** shows the representation of this reaction as directed bipartite graph

tite graphs, a modeling also used in Petri-net representations of pathways [8,21]. Here the reactions themselves are vertices and edges are binary relations connecting compounds of reactions with reaction vertices. Without loss of generality, reactions can be represented by directed bipartite graphs, see Fig. 2(b). This notation has the advantage that graph algorithms and standard graph drawing techniques can be used.

Our focus is on dynamic visualization in general and on pathway visualization in particular. *Dynamic visualization* is the generation of a diagram on demand at the time the drawing is needed. The placement of objects, e. g., substances or enzymes, and the routing of their connections is a typical graph drawing problem. However, the known graph drawing algorithms are inadequate for visualizing biochemical reaction networks according to the established conventions of biology and chemistry, see Fig. 3 for examples. Even combinations of standard graph drawing algorithms [1,13] are insufficient. Some solutions focus only on the placement of the main reactants and products [1], others place co-substances and enzymes in an unusual way like in PFBP [20]. Up to now, the best results have been obtained by using labeling techniques, where the enzymes and co-substances are considered as edge labels and placed separately [13,15]. Nevertheless, these algorithms can produce crossings between co-substances and edges. They also tend to clus-



(a) (b)

Fig. 3. Two visualizations of the same biochemical pathway **(a)** using a layered layout technique [23], and **(b)** using a force directed layout technique [5] (see also subsections 4.2 and 4.5 of the Technical Foundations Chapter). These drawings differ very much from typical drawings in biochemical textbooks

ter all co-reactants into one vertex and all co-products into another vertex. This is unacceptable for some reactions, especially for more complex reaction mechanisms as shown in Fig. 4. Furthermore, these drawings of reaction networks often contain many unnecessary edge crossings [13] which reduces their readability.

Interactive navigation through biochemical pathways is a particular advantage of electronic information systems over textbooks and posters. The concept of a hierarchy of reactions is a crucial feature when browsing through reaction networks on different levels of detail. Reactions should be separable into sub-reactions and combinable to pathways. Thus sub-reactions, reactions, and pathways should also be considered as reactions. In some systems this hierarchical organization is realized by linked web pages. The information in *KEGG* [10] is structured by web-links from pathway classes, e. g., carbohydrate metabolism or lipid metabolism, to single pathway diagrams. *UM-BBD* [3] provides a click-able overview picture and a list of pathways organized by functional groups. But it is also necessary to represent this hierarchy in the database. *EcoCyc* and *MetaCyc* [14] represent a hierarchy of reactions by linking pathway objects to objects representing the reactions.

Existing visualization techniques do not meet the conventions of biochemistry; they are far from the typical drawings in textbooks. Moreover, existing databases do not include all relevant information such as the hierarchy of reactions and layout constraints for specific pathways. These constraints are

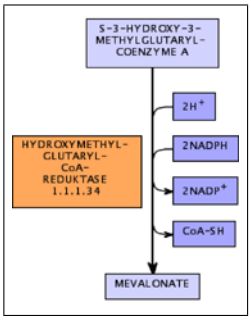


Fig. 4. The order of co-reactants and co-products is crucial for the understanding of the reaction mechanism and should be clearly visualized

necessary to distinguish pathways like the citrate cycle from arbitrary cycles in the network. Therefore, a new graph drawing algorithm and an adapted database schema are necessary.

The structure of this chapter is as follows. We start with a short overview of the *BioPath* system in Sect. 2. Section 3 deals with our visualization method. In subsection 3.1 we discuss requirements for the visualization of biochemical pathways. We then present a graph drawing algorithm which fulfils these requirements in subsection 3.2. Section 4 describes implementation aspects of the *BioPath* system, especially the application server in subsection 4.2 and the query engine in subsection 4.3. In Sect. 5 we show examples of the system.

2 Applications

The *Electronic Biochemical Pathways Project* [12] – a joint work of research groups at the universities of Erlangen, Mannheim, Passau and Spektrum Akademischer Verlag – provides convenient electronic access to the growing information of biochemical reactions at a high level of detail. An online exploration tool – *BioPath* [4] – demonstrates the quality of the data, the flexibility of the database schema, and the automatic visualization of pathways. It offers easy access to the information and progressive navigation through pathways. The database contains information about compounds and reactions, especially a hierarchy of reactions. A new graph drawing algorithm for the automatic visualization of pathways visualizes reaction networks according to the conventions of biochemistry. This algorithm and the *BioPath* system can be very useful for the analysis of biochemical pathways. *BioPath* explores all the advantages of an electronic version of the poster [16] over the printed one.

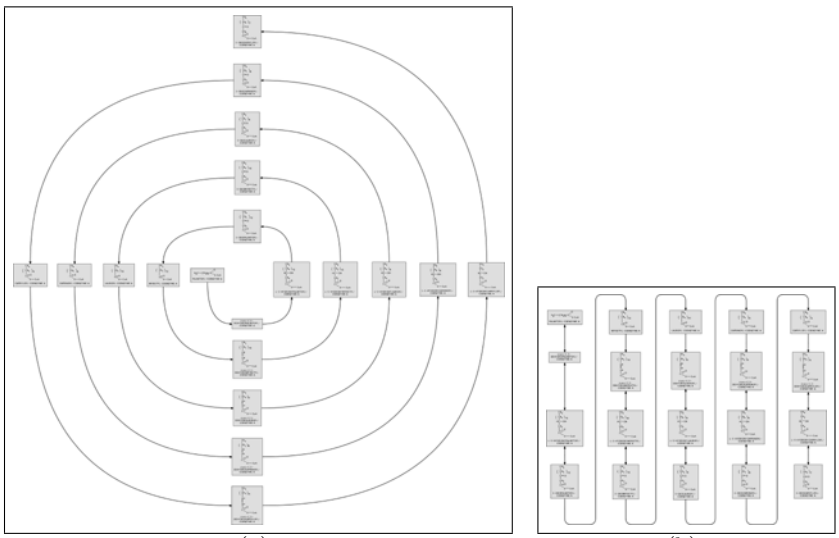
However, the field of applications for our algorithm is not limited to biochemical reactions. It can be employed wherever a Sugiyama style layout

as introduced in subsection 4.2 of the Technical Foundations Chapter is desired. Its comprehensive set of features described in Sect. 3 is of great value whenever a layout has to meet complex boundary conditions.

3 Algorithms

How shall we draw a biochemical pathway? What are the requirements for a good visualization algorithm? What criteria must be met? These are the crucial questions for the design of a solution that meets the needs of the users. It extends Knuth's question "How shall we draw a tree" to "How shall we draw a biochemical pathway". Knuth proposed to draw trees by layers and top-down, and this has become the common style in Computer Science. Drawing biochemical pathways is much harder, since they are directed hypergraphs. After a thorough analysis and intensive discussions with many experts we propose the following.

3.1 Visualization Requirements



(a) (b)

Fig. 5. Two drawings of an *open cycle* (degradation of fatty acids). For simplicity only the substances are shown. In **(a)** the typical visualization of this pathway as a spiral. In **(b)** the same pathway is shown in a more compact visualization that also emphasizes the corresponding reaction steps and substances

Compounds. The visualization of compounds of a reaction is user-specific. According to existing drawings in the literature, substances should be given by their name or their structural formula or both, co-substances should be displayed using their name or an abbreviation, and enzymes should be represented by their name or their classification number. In *BioPath* the conventions of the poster [16] are used. It displays the names and the structural formulas for substances and the names or abbreviations (e. g., ATP, NADP) for co-substances and enzymes.

The data for the compounds is retrieved from the database. It is transformed into vertices of the graph using varying vertex sizes. The size of the vertex provides the space needed to display the information of the object such as the name or the structural formula. As a consequence, the visualization algorithm must support different vertex sizes.

Reactions. A reaction is visualized by a reaction arrow from the reactants to the products. Enzymes and co-substances are placed on different sides beside this arrow. For co-substances their temporal order, which depends on the reaction mechanism, is important. Therefore they are placed according to this order. Overview diagrams often disregard enzymes and co-substances. Thus the visualization algorithm has to deal with reactions with and without enzymes and co-substances.

Reaction networks. As the temporal order of reactions in the network is crucial, the main direction of reactions should be clearly visible. In *BioPath* the main direction is from top to bottom. This is better than from left to right since it yields more compact representations and a better placement of objects with long textual information.

There is an exception from the top to bottom direction, which is used for the visualization of specific pathways such as the citrate cycle or the fatty acid synthesis. The structure of these cyclic reaction chains should be emphasized. These pathways are characterized by the continuous repetition of a reaction sequence in which the product of the sequence re-enters in the next loop as a reactant. There are two mechanisms. First, the reactant and the product of the reaction sequence are identical from loop to loop (e. g., citrate cycle). This is called a *closed cycle*. Second, the reactant of the reaction sequence varies slightly from the product (e. g., fatty acid cycle). This is called an *open cycle*.

Repetitions in cyclic structures must be clearly visible. In textbooks they are displayed as circles (closed cycles) or as spirals (open cycles). In a spiral equal reaction steps and corresponding substances are placed side by side to emphasis the cyclic structure. This drawing style has some drawbacks. It needs much space and it is difficult for a user to trace the reaction sequence, particularly in the outer part of the spiral; see Fig. 5(a). Furthermore, if only a part of the diagram can be shown on a restricted viewing area such as a

computer screen, extensive scrolling in all directions is necessary. Figure 5(b) shows the same pathway in a more compact visualization. The spiral has been unraveled and related reactions and substances have been aligned horizontally. This new drawing style avoids the above-mentioned disadvantages.

Sequences of reaction networks. Browsing through pathways can often be very useful for the study of reaction networks. One example is to start with an overview diagram and to refine a specific part of this diagram. Another mechanism of browsing is to add reactions to an existing reaction network interactively. In this case a sequence of pathway diagrams is generated. When a user knows a diagram of this sequence the next diagram is easier to understand if small changes between the successive reaction networks imply only small changes of the corresponding pathway diagrams. The drawing of the unchanged part of the network should be preserved, but this is not always possible. For example, there may not be enough space for the insertion of a new reaction. In this case the relative positions of the old objects should be preserved. In graph drawing this technique is called *preserving the mental map* [19].

3.2 The Graph Drawing Algorithm

We model reaction networks as directed bipartite graphs $G = (V_1 \cup V_2, E)$, where substances, co-substances, enzymes, and reactions are represented as vertices $v \in V = V_1 \cup V_2$ and their connections are represented as directed edges $e \in E$; see Fig. 2(b). The vertices of this bipartite graph are labeled by the name of the compounds, their role in the reaction (enzyme, substance, or co-substance), their involvement in open or closed cycles, and their rank. The labels are used for the generation of the image of the reaction or reaction network and are translated into associated text, colors, and layout constraints. A layout constraint describes a left-to-right or top-to-bottom relationship and means for example that an enzyme should appear to the left of the reaction arrow.

The visualization algorithm is based on the graph drawing algorithm by Sugiyama et al. [23] for the computation of layered layouts of directed acyclic graphs; see subsection 4.2 of the Technical Foundations Chapter. The main extensions of the algorithm are:

Clustering. *Clustering* means the grouping of disjoint subgraphs into new vertices. Each grouped subgraph can be drawn with its own layout algorithm. The size of the corresponding new vertex is determined by the space of the drawn subgraph. In this context the routing of edges from vertices of the subgraph to vertices outside the subgraph is a difficult problem.

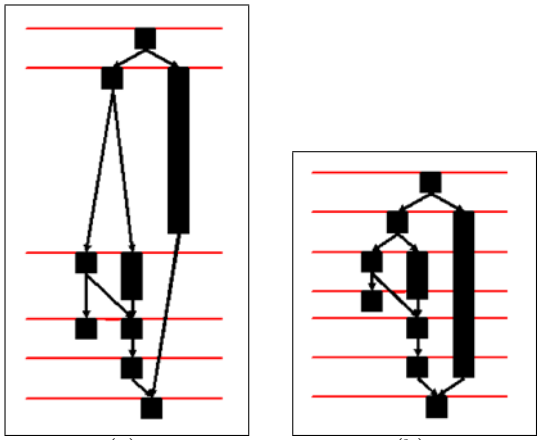


Fig. 6. Layering mechanisms: **(a)** the typical global layering; **(b)** the local layering. The local layering mechanism considers the vertex sizes correctly and leads to compact placements

Vertex sizes. The Sugiyama algorithm has only a coarse view of the size of vertices. It uses a global layering mechanism by placing all vertices in horizontal layers depending on their topological order. The distance between two layers is defined by the highest vertex of the above layer. Our algorithm is adapted to varying vertex sizes and is extended to a local layering mechanism where all vertices are placed in layers depending only on the placement of all predecessors; see Fig. 6.

Layout constraints. *Layout constraints* are additional application-specific requirements on the placement of objects. The layout algorithm considers the following types of layout constraints: *top-bottom* constraints to place one vertex below another one; *left-right* constraints for the horizontal order of two vertices; *horizontal* constraints to force the same layer for two vertices; and *vertical* constraints to force the same x-coordinate for two vertices. Layout constraints are used to draw open and closed cycles according to the specified requirements, and to preserve the mental map in related drawings. Algorithm 1 shows the main steps of the layout algorithm.

It is known that optimization problems connected to some steps are $\mathcal{NP}$-hard [6]; see subsection 4.2 of the Technical Foundations Chapter. Therefore we use heuristics for these steps. The time critical part of the complete algorithm is step 3 to step 7. Let G be a connected graph, n be the number of vertices and m be the number of edges of G. For the steps 3, 6 and 7 different heuristics with complexities from linear time to $\mathcal{O}(n^4)$ (for step 6) can be chosen, where a higher running time usually gives better results. However,

Algorithm 1: Main steps of the layout algorithm

Input : A directed graph (representing a reaction network) with vertex sizes, vertex types (substance, co-substance, enzyme, reaction) and an indication of open and closed cycles

Output: The drawing of the graph

1 Compute local layouts for co-substances and enzymes of each reaction and cluster these vertices into large vertices, see Fig. 7. These large vertices are called *reaction vertices*

2 Insert layout constraints and temporary vertices for open and closed cycles as shown in Fig. 8

3 Reverse some edges to remove all cycles from the graph under the restriction of the top-bottom constraints

```
foreach reversed edge do
    if possible (no new cycles occur by the following operations) then
        turn the local layout of the adjacent reaction vertex by 180 degree
        (this reaction goes now from bottom to top) and change the direction
        of all edges adjacent to this reaction vertex.
    else
        insert a vertex on each edge adjacent to the corresponding reaction
        vertex to allow additional bends for these edges.
    end
end
```

4 Assign vertices to horizontal layers under the restriction of horizontal constraints. All edges should be directed from top to bottom. For each vertex the distance to its predecessors should be minimized. This step computes the y-coordinates of vertices

5 Compute a proper layering by inserting temporary vertices for long spanning edges and long spanning vertices, see Fig. 9

6 Permute the order of vertices within each layer to reduce the number of edge crossings in the layered graph such that the left-right and the vertical constraints are fulfilled

7 Compute x-coordinates for the vertices without changing the pre-computed order in the layers and under consideration of the vertical constraints

8 De-cluster reaction vertices, compute an edge routing by using the temporary vertices as additional base points for edges

9 (Additional) Insert constraints for preserving the mental map in the next drawing

most important are steps 4 and 5, because during these steps up to $n \cdot m$ temporary vertices and edges will be inserted in the graph. This influences the running time of the subsequent steps of the algorithm. The layout algorithm has been tested with more than 200 graphs. These graphs represent reaction networks with up to 50 reactions (or correspondingly up to 300 vertices). Using a standard PC we were able to layout these graphs within a few seconds using the heuristics with the higher complexities.

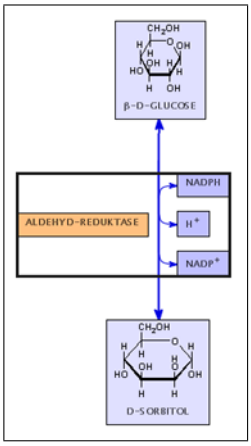


Fig. 7. Clustering of co-substances and enzymes of each reaction into a large vertex (reaction vertex). The positions of the co-substances and enzymes are computed separately. Enzymes are placed to the left of the reaction arrow, co-substances to the right according to their order. This determines the size of the reaction vertex. In the subsequent steps of the layout algorithm the reaction vertex is used instead of the clustered vertices. The order of the co-substances and their current role (co-reactant or co-product) are retrieved from the database

4 Implementation

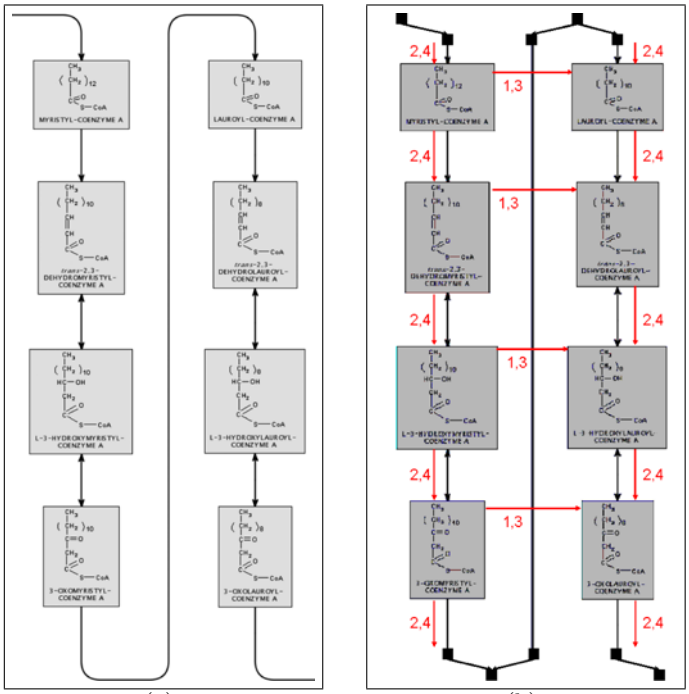
4.1 Overview

BioPath is a classical 3-tier web application. See Fig. 10 for an architecture overview.

Client Tier. Users access the *BioPath* service using a web browser like Netscape Navigator or Microsoft Internet Explorer. They enter their queries into HTML forms. The browser passes the query data to the *BioPath* application server by sending a HTTP request. When the application server has finished query processing, the browser displays the returned query results. Clicking on pathway images or on internal links triggers another HTTP request.

Application Tier. The main part of *BioPath* is the application server. It accepts queries, retrieves the corresponding data from the database, computes the result and delivers it to the client tier.

Data Storage Tier. The data of *BioPath* is stored in a relational database management system (currently IBM DB2) using an object oriented schema [11].



(a) (b)

Fig. 8. Image **(a)** shows the visualization of a part of an open cycle (two loops of the pathway). The diagram **(b)** shows the layout constraints to receive this layout: 1 - Horizontal constraints between corresponding substances in different loops of the pathway. 2 - Top-bottom constraints between consecutive vertices of each loop of the pathway (the vertex at the end of the arrow should be placed below the vertex at the begin of the arrow). 3 - Left-right constraints between vertices in the same layer in neighboring loops. 4 - Vertical constraints between consecutive vertices of each loop to force a vertical placement of vertices. For technical reasons some additional vertices are necessary. For closed cycles similar constraints are used

4.2 The Application Server

The heart of the *BioPath* system is the application server. It consists of several components, some of which are implemented in Java, some in C++ for execution speed. The communication between the Java and C++ components is done via the Java Native Interface (JNI). The application server has been developed mainly on Linux and Windows in an operating system independent manner, and therefore should be easily portable to any platform for which Java and a recent C++ compiler are available. It contains the following parts.

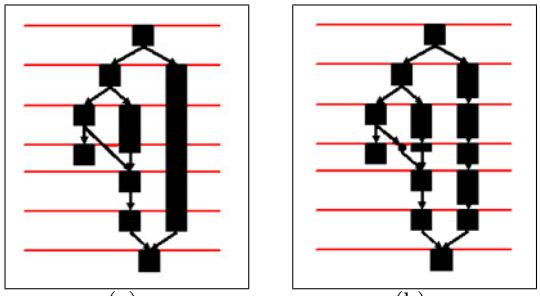


Fig. 9. Image **(a)** shows the assignment of vertices to horizontal layers. Long spanning edges are edges that cross at least one layer, long spanning vertices belong to at least two layers. In image **(b)** these edges and vertices are replaced by chains of temporary vertices

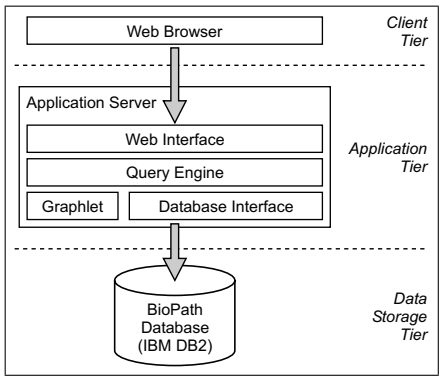


Fig. 10. The *BioPath* system architecture.

Web Interface. The web interface is responsible for the communication with the client tier. It receives the query in terms of a HTTP request with associated parameters. It parses the request and triggers the corresponding functionality of the query engine which processes the query and returns the result as a graph. The web interface uses the layout and graphics engine to transform the graph into a picture and delivers it to the client as a GIF or PNG image with a corresponding image map and a HTML page. The web interface is implemented in Java based on the Java Servlet Technology.

Query Engine. The query engine executes the user queries. It extracts the required information from the database and builds a pathway, represented as a graph with attributed vertices and edges. A description is given in subsection 4.3. The query engine is implemented in C++ using the "Graph

Template Library" (GTL), a C++ library providing a data structure for graphs.

Database Interface. The communication between the query engine and the database is done via the database interface. This encapsulation simplifies adapting *BioPath* to other data sources. The database interface is written in C++.

Layout Engine. The graphs generated by the query engine do not have a geometric representation. For the display an image of the graph must be computed. As a prerequisite, coordinates for the vertices and edges must be calculated. This is done by the layout engine. It uses the graph drawing algorithm described in Sect. 3 to compute a layout of the biochemical network. The layout engine is implemented in C++ using Graphlet [7].

Graphics Engine. The graphics engine generates images and image maps from the attributed graphs computed by the layout engine. It is implemented in Java using the Java interface of GTL and Graphlet.

4.3 Query engine

The query engine processes the queries from the web interface, issues database queries, transforms the result and returns the answer to the web interface. The *BioPath* database features an elaborate schema designed by Kanne [11]. It consists of about 115 entities and 70 relations. We concentrate only on the part relevant to the query process.

The biochemical reactions are the main components of the database schema. Each reaction has a set of attributes including the participants, i. e., the substances involved in the reaction. These are classified into enzymes, reactants and products. Reactants and products are further classified into primary and non-primary. Primary reactants and products are used to compose pathways. Non-primary participants have a rank which specifies their relative order in the reaction and is used by the layout engine. The graph representation of the data is described in Sect. 3.

There are several types of queries. A search query on substances is a standard query for information on a substance, which is passed on directly to the database. The answer is a text or a structural formula as it is stored in the database.

A query on reactions and reaction nets shall return an image. Then the task of the query engine is building a reaction graph. A search-reaction query first collects all participants of the reaction from the database. For each participant the query engine creates a vertex and a vertex for the reaction itself, and labels the vertices with the name of the substance, the role and the rank.

Then it adds directed edges between the reaction vertex and the participant vertices. These are directed from the reactant vertices to the reaction vertex and from there to the product vertices. Reactions may be bi-directional, but their internal representation is always one-directional. To show bi-directional edges in the final drawing arrows on both ends of the edge are used.

The search of a reaction net from a source substance to a target substance is finding a sequence of reactions of a predefined maximal length. In *BioPath* two reactions are combined, if a primary product of the first is a primary reactant of the second. This is the common notion of connectivity in such graphs. *BioPath* uses a recursive query to build all sequences starting from the source substance up to a predefined length. It then selects those reaching the target substance. In more detail, this query returns a collection of reactions, which are transformed into labeled graphs as described above. Vertices of primary participants are identified, if they represent the same substance. The so obtained directed graph is returned to the web interface.

5 Examples

BioPath is an online system for the retrieval of information on substances and biochemical pathways. It offers several ways to explore biochemical data and displays the data in the common style as text or graphics.

5.1 Start Page

The start page is shown in Fig. 11 and presents general information about the *BioPath* project and the partners, a short introduction, a user guide and a starting point for the exploration of the biochemical data.

The user guide introduces into *BioPath* and explains how to find information about substances, enzymes, pathways, and reactions. It illustrates the visualization of reactions and pathways and displays the main features of *BioPath*.

5.2 Overview Diagram

The overview diagram in Fig. 12 has been derived from the overview diagram of the *Biochemical Pathways* atlas [17]. It shows a selection of important biochemical pathways. All colored substances and reactions in the diagram can be clicked to receive more information. This is the only manually made reaction diagram in *BioPath* to ensure similarity to the overview diagram of the atlas.

5.3 Search Substances

Substances can be searched by their name or a part of it, see Fig. 13. The result shows biochemical data such as structural and empirical formula, weight,

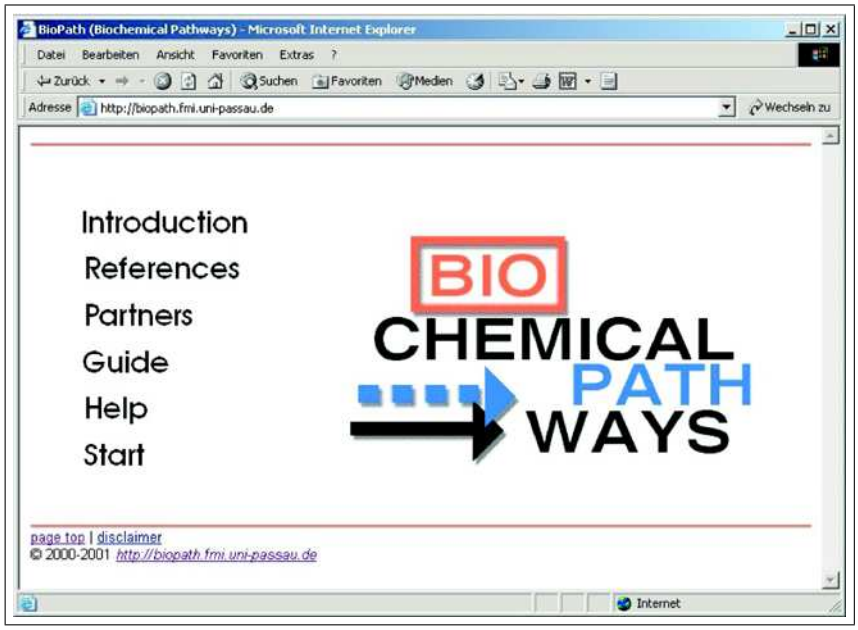


Fig. 11. Start page of *BioPath*

enzyme classification number and charge. Additionally all reactions in which the substance participates as a reactant, a product or an enzyme are listed. Fig. 14 is an example for Chlorophyll.

5.4 Search Pathways and Reactions

As shown in Fig. 15, pathways can be searched by their name or a part of it. *BioPath* computes a graphical representation containing information like participating enzymes and substances of the reaction or pathway and also shows all pathways the reaction belongs to. As in the overview diagram, displayed substances and reactions can be clicked to receive detailed information.

5.5 Search reaction net

The most advanced feature of *BioPath* is the search for a reaction net between two substances. The query panel is shown in Fig. 16. *BioPath* searches all sequences of reactions between two substances up to a given length. Figure 17 shows the result of the query "Search the reaction net between *Maltose* and *D-Gluconate* up to length 10". The search can be restricted to a single pathway and in depth. The later is also used to speed-up the search time and database queries. The depth counts the number of intermediate substances.

The usability of *BioPath* is limited by the given data on substances, reactions, and reaction nets, which is only an excerpt of the information on

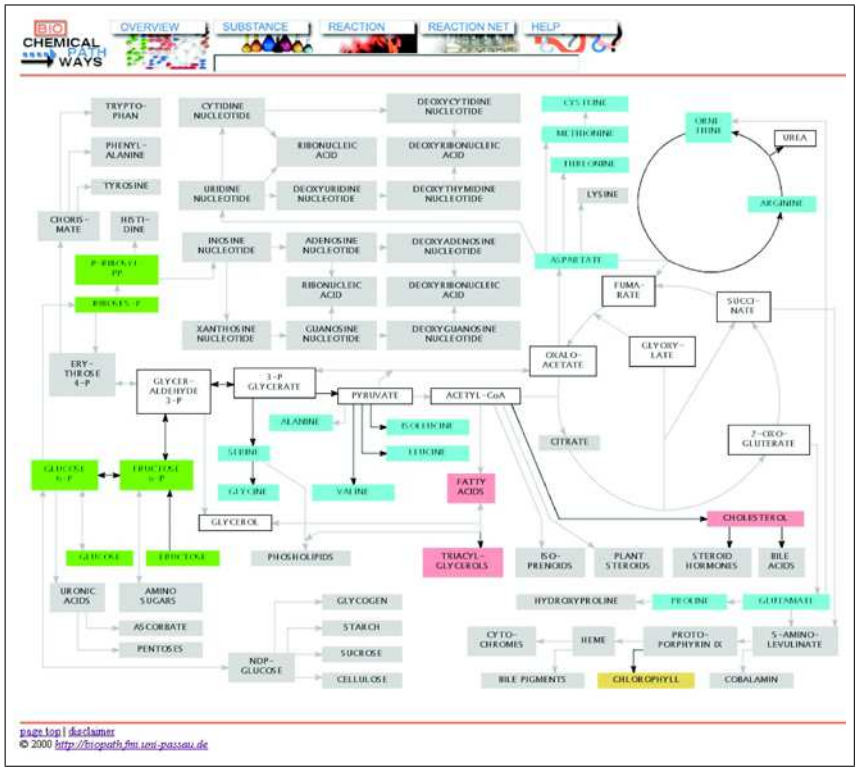


Fig. 12. The overview diagram is a starting point for exploration of the pathways

Fig. 13. Searching a substance

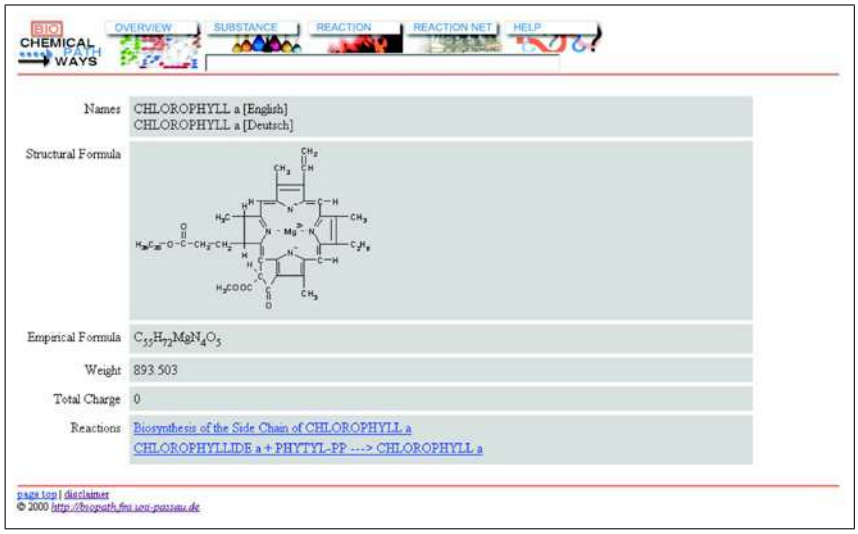


Fig. 14. Information about Chlorophyll

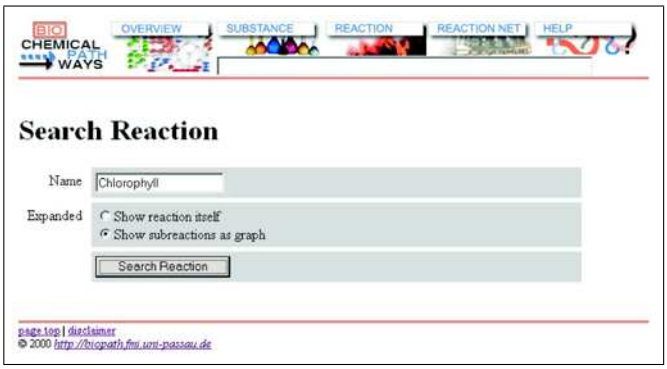


Fig. 15. Searching for the biosynthesis of the Side Chain of Chlorophyll a

the poster or in the atlas or in commercial databases. However, this does not restrict our general approach.

6 Software

BioPath is accessible via `http://biopath.fmi.uni-passau.de`.

References

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

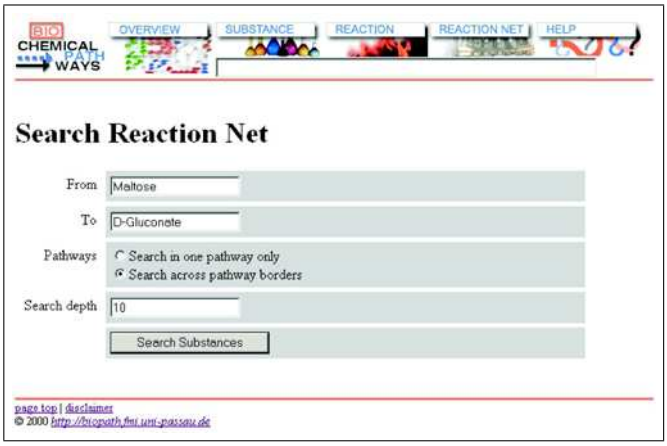


Fig. 16. Searching a reaction net

2. F. J. Brandenburg, B. Gruber, M. Himsolt, and F. Schreiber. Automatische Visualisierung biochemischer Information. In R. Hofestädt, editor, *Proceedings of the Workshop Molekulare Bioinformatik, GI Jahrestagung*, pages 24–38. Shaker Verlag, 1998.
3. L. B. Ellis, C. D. Hershberger, and L. P. Wackett. The University of Minnesota Biocatalysis/Biodegradation Database: Microorganisms, Genomics and Prediction. *Nucleic Acids Research*, 28(1):377–379, 2000.
4. M. Forster, A. Pick, M. Raitner, F. Schreiber, and F. J. Brandenburg. The System Architecture of the BioPath System. *In Silico Biology*, 2(3):415–426, 2002.
5. T. Fruchterman and E. Reingold. Graph Drawing by Force-directed Placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.
6. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.
7. M. Himsolt. Graphlet: Design and Implementation of a Graph Editor. *Software - Practice and Experience*, 30(11):1303–1324, 2000.
8. R. Hofestädt and S. Thelen. Qualitative Modeling of Biochemical Networks. *In Silico Biology*, 1:39–53, 1998.
9. International Union of Biochemistry and Moleculare Biology, Nomenclature Commitee. *Enzyme Nomenclature*. Academic Press, 1992.
10. M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acid Research*, 28(1):27–30, 2000.
11. C.-C. Kanne. BioPath database scheme. Internal documentation, 2000.
12. C.-C. Kanne, F. Schreiber, and D. Trümbach. Electronic Biochemical Pathways. In *Proceedings 7th Symposium on Graph Drawing (GD'99)*, volume 1731 of *LNCS*, pages 418–419. Springer Verlag, 1999.
13. P. D. Karp and S. M. Paley. Automated Drawing of Metabolic Pathways. In H. Lim, C. Cantor, and R. Bobbins, editors, *Proceedings of the 3rd International Conference on Bioinformatics and Genome Research*, pages 225–238, 1994.

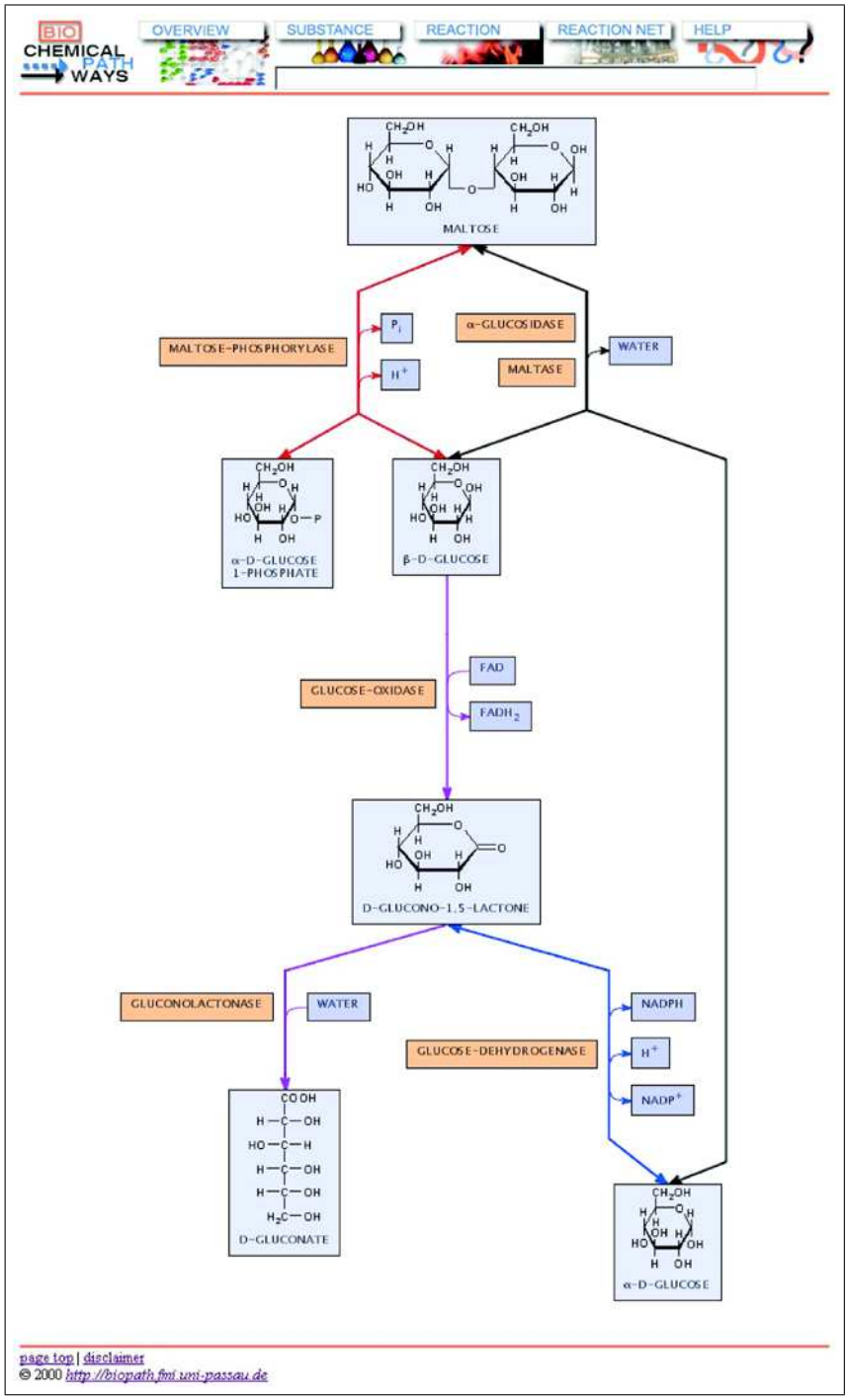


Fig. 17. A reaction net. This view contains structural formulas for substances. The image is automatically produced by the layout algorithm of *BioPath*

14. P. D. Karp, M. Riley, M. Saier, I. T. Paulsen, S. M. Paley, and A. Pellegrini-Toole. The EcoCyc and MetaCyc database. *Nucleid Acids Research*, 28:56–59, 2000.
15. P. Mendes. Advanced Visualization of Metabolic Pathways in PathDB. In *Proceedings of the 8th Conference on Plant and Animal Genome*, 2000.
16. G. Michal. *Biochemical Pathways (Poster)*. Boehringer Mannheim, Penzberg, 1993.
17. G. Michal. *Biochemical Pathways*. Spektrum Akademischer Verlag, Heidelberg, 1999.
18. G. Michal, 2000. Personal Communication.
19. K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout Adjustment and the Mental Map. *J. of Visual Languages and Computing*, 6:183–210, 1995.
20. `http://www.ebi.ac.uk/research/pfmp/`.
21. V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman. Petri Net Representations of Metabolic Pathways. In L. Hunter, D. Searls, and J. Shavlik, editors, *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB'93)*, pages 328–336, 1993.
22. L. Stryer. *Biochemie*. Spektrum Akademischer Verlag, Heidelberg, 1995.
23. K. Sugiyama, S. Tagawa, and M. Toda. Methods for Visual Understanding of Hierarchical System Structures. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(2):109–125, 1981.

Index