Université Libre de Bruxelles
Faculté des Sciences - Biologie moléculaire
Service de Conformation des Macromolécules Biologiques et de Bioinformatique
(SCMBB)

PathwayBuilder: Metabolic pathway inference given a set of seed nodes

Karoline Faust

Mémoire présenté en vue de l'obtention du DEA en Bioinformatique

Août 2006

Promoteur: Prof. Jacques van Helden

Acknowledgements

First of all I would like to thank my supervisor Jacques van Helden, who gave me this challenging topic, who encouraged and supported me throughout this work and who could always find time for explanations and discussions when I needed them despite of his overfull timetable.

A big thanks as well to Didier Croes, on whose research this final work is based. He was always open for questions and came several times back to the lab to discuss and to help me.

This work owes a lot to the aMAZE team, namely Christian Lemer, Fabian Couche, Frédéric Fays, Hassan Anerhour, Olivier Hubaut and Simon De Keyzer. They introduced me to Eclipse and Java and answered questions about aMAZE, graph algorithms and the good practice of programming. Fabian gave me direct access to his backtracking algorithm and Christian taught me how to use Ant and had useful suggestions to solve many problems.

In addition, I would like to thank the SCMBB members (Raphaël Leplae, Gipsi Lima Mendez, Rekin's Janky, Raul Mendez, Marc Lensink, Olivier Sand, Jean-Valéry Turatsinze, Sylvain Brohée, Benoît Dessailly, Ariane Toussaint and Fernanda L. Sirota) who welcomed me very friendly in the lab and who helped me whenever I had computer-related or other problems.

Thanks as well to Grégoire Dooms and Stéphane Zampelli, who introduced me to (CP)Graph, to Jean-Noël Monette, Pierre Schaus and their supervisor Pierre Dupont, who implemented a number of graph algorithms I plan to use and to Yves Deville, who is together with Jacques van Helden scientific director of the TransMAZE and BioMAZE projects, from which this work benefitted.

Finally, I thank Didier Gonze, who cheered me up whenever I was despairing.

Summary

Co-expressed genes often code for proteins that are involved in a common biological function such as a metabolic pathway. Tools that can detect metabolic pathways based on groups of co-expressed genes would be of value for the interpretation of microarray data.

The aim of this final work was to develop such a tool, the PathwayBuilder, which receives a number of input enzymes and connects best the reactions they catalyze in a given metabolic graph under the given constraints. The output is a graph representing the inferred metabolic pathway. This approach differs from pathway mapping, because it allows new combinations of known reactions and compounds to occur.

The PathwayBuilder requires seeds (compounds, reactions or groups of reactions) and a metabolic graph to perform pathway inference. To obtain the seeds from a group of co-expressed genes, the enzymes among them need to be identified and linked to their reactions. This is achieved by using the EC number annotation of the input enzymes. There are different ways to represent metabolic data in graphs. In this work, compounds and reactions from KEGG have been collected into a directed, bipartite and weighted graph. Ubiquitous compounds are avoided by using a new approach introduced by Didier Croes, which relies on compound connectivity.

The novelty of the PathwayBuilder in comparison to previous pathway inference tools is that it can handle a set of seeds. Pathway inference with two seeds is successful for most pathways [CROES ET AL. 05], but fails for some as for example the purine biosynthesis pathway in *E. coli*. For this reason this pathway was chosen to test the PathwayBuilder. It could be shown that additional seed nodes increased the accuracy of its inference.

The next step will be to improve and to validate the PathwayBuilder on a number of annotated pathways before applying it to microarray data.

Definitions

A number of definitions important for this work are given below. In brackets, alternative names for the defined terms are listed.

Graph

A graph G is a finite nonempty set of objects called vertices *(or nodes)* together with a (possibly empty) set of unordered pairs of distinct vertices of G called edges. (cited from [CHARTRAND & LESNIAK 96], words in italics added by the author)

Directed graph (digraph)

A directed graph or digraph D is a finite nonempty set of objects called vertices *(or nodes)* together with a (possibly empty) set of ordered pairs of distinct vertices of D called arcs or directed edges. (cited from [CHARTRAND & LESNIAK 96])

Bipartite graph

A graph is k-partite, $k \geq 1$, if it is possible to partition V(G) *(the set of vertices of graph G)* into k subsets $V_1, V_2, ..., V_k$, such that every element of E(G) *(the set of edges of graph G)* joins a vertex of V_i to a vertex of V_j, $i \neq j$. For $k = 2$, such graphs are called bipartite graphs. (cited from [CHARTRAND & LESNIAK 96], words in italics added by the author)

Connectivity (degree)

The outdegree of a vertice v of a digraph D is the number of arcs of D that are adjacent from v. The indegree of v is the number of arcs of D adjacent to v. The degree of a vertice v in D is defined as the sum of its indegree and outdegree. (after [CHARTRAND & LESNIAK 96])

Neighborhood

The neighborhood $N(v)$ of a vertex v in a graph G is the set of all vertices of G that are adjacent to v. (cited from [CHARTRAND & LESNIAK 96]). An element of $N(v)$ is called neighbor node or neighbor.

Path

For two nodes u and v in a graph G, a $u - v$ walk of G is a finite, alternating sequence of nodes and edges, beginning with node u and ending with node v. A $u - v$ path is a $u - v$ walk in which no node is repeated. (after [CHARTRAND & LESNIAK 96])

Metabolic graph

The metabolic graph is defined in this work as a connected, bipartite, directed graph consisting of compounds and reactions.

Metabolic pathway (pathway)

A metabolic pathway is a subgraph of the metabolic graph. Sometimes, this term is abbreviated to pathway. Metabolic pathways described in textbooks or metabolic databases are referred to as annotated or reference pathways.

Seed node (seed)

A seed node is an element of the compound node set or the reaction node set of the metabolic graph. It is given as input to the PathwayBuilder.

Shortest path problem

In graph theory, the shortest path problem is the problem of finding a path between two vertices such that the sum of the weights of its constituent edges is minimized. (after: http://en.wikipedia.org/wiki/Shortest_path_problem)

K shortest path problem

The k shortest paths problem is to list the k paths connecting a given source-destination pair in the digraph with minimum total length. (cited from [Eppstein 94]) A k shortest path algorithm is an algorithm that solves the k shortest path problem.

EC number

A systematic classification of enzymes was introduced by the Enzyme Commission 1961 and is based on the chemical reactions they catalyze. The assignment of code numbers (EC numbers) to enzymes follows this classification system. EC numbers contain four elements, separated by points, with the following meaning:

1. The first number shows to which of the six main divisions (classes) the enzyme belongs.
2. The second figure indicates the subclass.
3. The third figure gives the sub-subclass.
4. The fourth figure is the serial number of the enzyme in its sub-subclass.

(after: Nomenclature Committee of the International Union of Biochemistry and Molecular Biology, http://www.chem.qmul.ac.uk/iubmb/enzyme/rules.html)

Abbreviations

E. coli	*Escherichia coli*
FN	False Negative
FP	False Positive
GDL	GraphDataLinker
IQL	Igloo Query Language
KEGG	Kyoto Encyclopedia of Genes and Genomes
PGDB	Pathway/Genome DataBase
S. cerevisiae	*Saccharomyces cerevisiae*
TN	True Negative
TP	True Positive

Contents

1 Introduction

The development of high-throughput techniques such as microarrays enabled monitoring of gene expressions at a genomic scale. Thus, large amounts of data are produced that need processing and interpretation to derive hypotheses from them.

One approach commonly applied to the interpretation of microarray data has been termed "Guilt by association" [Quackenbush 03]. It states that co-expressed genes (genes whose expression values are either increased simultaneously or decreased simultaneously with respect to a reference) are likely to contribute to a common biological function such as a pathway. Given the validity of this approach, microarray data could be used to infer metabolic pathways from groups of co-expressed genes.

1.1 Summary of the project

1.1.1 Context of the final work

This work is a preparation to my PhD subject, which consists of the inference of metabolic pathways from sets of co-expressed genes obtained from microarray experiments. This work started in January 2006. During this period, I focused on the extension of an existing k-shortest path finding algorithm (backtracking with constraints) to accept multiple seeds as input. I implemented a first version of the PathwayBuilder and applied it to a few study cases.

1.1.2 Goal

The goal of the final work is to develop a tool, termed PathwayBuilder, which performs the inference of metabolic pathways given a metabolic graph and a set of enzyme-coding genes.
Thus, this work strives to answer the following question: Given a set of co-expressed genes, which pathway(s) could the enzymes they code catalyze together?

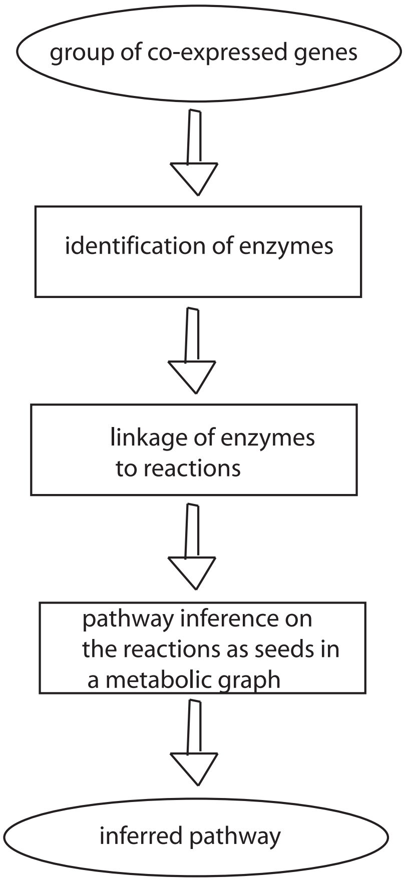

group of co-expressed genes
identification of enzymes
linkage of enzymes to reactions
pathway inference on the reactions as seeds in a metabolic graph
inferred pathway

Figure 1.1: Summary of the strategy that is applied to infer metabolic pathways from groups of co-expressed genes.

1.1.3 Strategy

The task can be divided into the following steps: first, the co-expressed genes are obtained from the microarray data set. Next, the enzymes among those genes are identified using available annotation. Then, an extension of a k shortest path algorithm, the PathwayBuilder, is used to infer pathways from the reactions catalyzed by those enzymes. The PathwayBuilder returns a pathway that connects as many of the given seed reactions as possible under the given criteria. Figure 1.1 summarizes the procedure.

1.1.4 Applications

The main application of the PathwayBuilder will be to ease the interpretation of microarray data with respect to metabolic pathways. In contrast to available tools it does not rely on a matching of the sets of co-expressed enzymes to pre-defined pathways, but infers pathways from a metabolic graph. Those inferred pathways might be identical to known pathways, they might be variants of known pathways or they might even be unknown pathways composed of known reactions and compounds.

In addition, the PathwayBuilder could be used in metabolic engineering to suggest possible pathways for the biosynthesis of a desired compound or the biodegradation of an unwanted compound. With respect to the prediction of biodegradation, the ability of the PathwayBuilder to merge metabolic information from several organisms is particularly useful, since usually more than one organism is involved in biodegradation pathways.

Furthermore, the PathwayBuilder could be extended to avoid solution paths containing given compounds or reactions. If pathway inference is then performed, alternative pathways without those compounds/reactions might be found. These alternative pathways might help to predict or explain the effects of enzyme knockouts on the metabolism of the investigated organism.

1.2 Representation of metabolic data

Success of pathway inference depends on metabolic databases and the choice of representation of metabolic data. In the following, two metabolic databases important for this work a shortly presented and different ways of metabolic data representation as graphs are summarized.

1.2.1 Representation of metabolic data in databases

A large number of metabolic databases has been published in recent years (a comparison of a selection of them is given in [Wittig & de Beuckelaer 01]). They can be divided into two different categories according to Karp [Karp 01]: Metabolic pathway databases and metabolic pathway/genome databases (PGDBs). The former describe metabolic pathways, their reactions, enzymes and compounds, whereas the latter integrate genomic information with pathway information. Thus, PGDBs allow to link enzymes to the reactions they catalyze, which is crucial for the current work.

Two PGDBs (more strictly: collections of PGDBs for several organisms) are relevant for this work, namely KEGG [Kanehisa et al. 02] and BioCyc [Karp et al. 05]. KEGG stores metabolic pathways in form of maps. The maps show merged metabolic data from all annotated organisms. On the maps, compound names and clickable EC numbers are displayed, which give details on the enzyme(s) that can perform the chemical reaction described by the EC number and list the reactions associated to this EC number. If one organism is selected from a list of organisms, organism-specific EC numbers on the maps are colored in green. If no organism is selected, a map without highlighted EC numbers is shown, which is also called reference map. A query form allows searching for objects (gene names, KEGG reaction ids, EC numbers or KEGG compound ids) in the maps.

In contrast to KEGG, BioCyc stores a number of annotated organism-specific metabolic pathways. These pathways can be queried for gene names, RNA names, protein names, pathway names, EC numbers, reaction names and compound names. In addition, BioCyc gives literature references for and comments on the pathways. Its pathway display is more flexible than the KEGG maps (different degrees of detail) and its pathway tools allow cross-species comparison.

Both databases are based on different annotation efforts. Thus, they can both be used for independent validation of the PathwayBuilder.

1.2.2 Representation of metabolic data as graphs

In biochemistry textbooks, metabolic pathways are usually depicted as a sequence of reactions with their main educts (substrates) and products connected by arrows, which represent the direction of the reaction. The arrows are often labeled with the name of the enzyme catalyzing the reaction. Often, side educts and products (compounds that do not take part in the next reaction) are included and formatted in a different way than the main educts and products.

Maps have been published that link these pathways to a network, thereby providing an overview of known metabolism (for example the Roche Applied Science "Biochemical Pathways" wall chart, compiled by Gerhard Michal, digital version available at http://ca.expasy.org/tools/pathways/).

From this wall chart it can be seen that borders between pathways cannot easily be defined. Pathways do not necessarily reflect biological units and the definition of some is arbitrary. A representation of metabolism as a network is therefore more appropriate for pathway analysis and inference. Graphs have been widely used in the literature to achieve this.

In Table 1.1 an (incomplete) overview on different representations of metabolic networks as graphs in the literature is given.

Authors	Nodes	Arcs	Directed	Bipartite	Weighted	Treatment of ubiquitous compounds	Remarks
[Fell & Wagner 00]	compounds	reactions	no	no	no	exclusion	small world property stated for *E. coli* metabolic network
[Jeong et al. 00]	set 1: compounds set 2: reactions	educt-reaction and reaction-product relationships	yes	yes	no	none	scale freeness of small world property stated for metabolic networks
[Küffner et al. 00]	places: compounds, transitions: reactions	connecting places with transitions and vice versa	yes	yes	no	exclusion by constraints	constrained pathway enumeration on Petri nets
[Forst & Schulten 01]	reactions	compounds	no	no	yes	none	distances of metabolic networks based on sequence information calculated

Authors	Nodes	Arcs	Directed	Bipartite	Weighted	Treatment of ubiquitous compounds	Remarks
[VAN HELDEN ET AL.01]	set 1: compounds, set 2: reactions	educt-reaction and reaction-product relationships	yes	yes	no	exclusion	pathway inference given a set of seed nodes
[GOESMANN ET AL.02]	compounds	reactions	yes	no	no	avoided by using annotated pathways	pathway analysis
[SIRAVA ET AL.02]	set 1: compounds, set 2: reactions	educt-reaction and reaction-product relationships	yes	yes	yes	exclusion	metabolic pathfinding tool for newly sequenced organisms
[MCSHAN ET AL.03]	compound lists	rules (state transitions)	yes	no	no	rule-based	metabolic pathfinding tool
[RAHMAN ET AL.04]	compounds	reactions	yes	no	no	chemical similarity	metabolic pathfinding tool
[ARITA 04] [ARITA 03]	small compounds (metabolites)	atom mappings	yes	no	yes	atom tracing	small world property for *E. coli* metabolic network rejected
[HOU ET AL.04]	compound lists	rules	yes	no	no	rule-based	prediction of biodegradation

Authors	Nodes	Arcs	Directed	Bipartite	Weighted	Treatment of ubiquitous compounds	Remarks
[CROES ET AL.05] [CROES ET AL.06]	set 1: compounds, set 2: reactions	educt-reaction and reaction-product relationships	yes	yes	yes	exclusion by weight	metabolic pathfinding tool

Table 1.1: The table summarizes some metabolic graph structures used in the literature.

These representations can be divided in different categories with respect to their treatment of compounds and reactions: The first category includes graphs that have reactions as nodes linked by arcs that represent compounds shared by reactions [FORST & SCHULTEN 01]. The second (and more common) strategy is to represent compounds as nodes and reactions as arcs [FELL & WAGNER 00], [GOESMANN ET AL.02], [RAHMAN ET AL.04]. A third way is to use bipartite graphs that represent compounds and reactions as nodes and educt-reaction/reaction-product relationships as arcs [KÜFFNER ET AL.00], [VAN HELDEN ET AL.01], [SIRAVA ET AL.02], [CROES ET AL.05].

It is also interesting to summarize the different strategies, which are used to avoid ubiquitous compounds. Ubiquitous compounds are small compounds that appear in a large number of reactions, either as co-factors or side compounds. Despite their high connectivity, they can generally not be considered as valid intermediates between reactions in a metabolic pathway. Consider for example the two following reactions (KEGG reaction ids are given in brackets):
Lactose + H2O $\Longleftrightarrow$ alpha-D-Glucose + D-Galactose (R01678) and
L-Tryptophan + H2O $\Longleftrightarrow$ Indole + Pyruvate + NH3 (R00673).
Connected in a metabolic graph these two reactions would allow the following path, if ubiquitous compounds are not avoided:
alpha-D-Glucose $\Longrightarrow$ R01678 $\Longrightarrow$ H2O $\Longrightarrow$ R00673 $\Longrightarrow$ Indole.
This path gives the impression that alpha-D-glucose can be transformed into Indole in two steps, using H_2O as intermediate. This is however completely irrelevant for a biochemist. Four strategies have been proposed to tackle the problem of ubiquitous compounds: The first is to avoid ubiquitous compounds by excluding them from the graph [FELL & WAGNER 00], [VAN HELDEN ET AL.01], [SIRAVA ET AL.02]. This requires a list of ubiquitous compounds to be excluded. Ubiquitous compounds can be

excluded based on their connectivity, or based on a combination between connectivity and biochemical properties [van Helden et al. 00], [van Helden et al. 02]. It is however difficult to find the correct cut-off that divides ubiquitous from other compounds. Another strategy exploits knowledge of the molecular structure of compounds (atom tracing [Arita 04], chemical similarity [Rahman et al. 04]) to differentiate between main and side compounds. This strategy restricts the compound set of the metabolic graph to compounds whose structure is known.

A third strategy relies on rules to avoid short cuts via ubiquitous compounds. Rules are coded either as predecessor/successor lists for compounds [McShan et al. 03] or as allowed transformations for functional groups of compounds [Hou et al. 04]. This strategy requires an annotation effort to generate those rules.

Recently, a new strategy has been introduced by Didier Croes ([Croes et al. 05] and [Croes et al. 06]) that uses weighted metabolic graphs to avoid ubiquitous compounds. This strategy is presented in more detail in section 1.5.

1.3 Properties of metabolic graphs

The representation of metabolic data as graphs allows the application of graph analysis tools and concepts on them. These concepts include mathematical definitions for measurements such as the network diameter and the distribution of connectivity. Properties of metabolic graphs that have been stated based on these measurements will be described in more detail below.

Power law of connectivity

[Jeong et al. 00]

Jeong et al. investigated the topological properties of metabolic networks from 43 different organisms. They showed that the distribution of connectivity follows a power-law with negative exponent. This means that there are many nodes with low connectivity and some nodes, also termed hubs, which have a high connectivity.

Small world and scale-freeness

[Jeong et al. 00]

The small-world property is measured with the network diameter, which is defined as the length of the shortest path between any two nodes, averaged over all nodes of the network. Jeong and co-workers found that the network diameter for metabolic networks from 43 different organisms is around 3. This means that in average 3 steps are sufficient to go from any node to any other node in the network. The authors state that the small network diameter is a consequence of the highly connected hub nodes. Because the network diameter is the same for metabolic networks of different size, Jeong and co-workers regard metabolic networks as scale-free with respect to their small-world property.

Modularity

[Ravasz et al. 02]
This property is measured by the clustering coefficient, which is defined for a given node as the number of direct links between its neighbors divided by the number of possible links between the neighbors. The clustering coefficient of the graph is the averaged clustering coefficient of all its nodes. The topological overlap, defined by Ravasz and co-workers as the number of nodes to which two selected nodes are both linked, is another measurement of modularity. Ravasz et al. applied these two measurements to a number of metabolic networks from different organisms and concluded that metabolic networks are modular.

Hierarchical architecture

[Ravasz et al. 02]
Ravasz and co-workers state that having both, modularity and scale-free organization of metabolic networks, poses a problem, because scale-freeness strongly restricts modularity. To solve this conflict, Ravasz and co-workers propose that metabolic networks have a hierarchic topology. To show that this is indeed the case, they measured the distribution of the clustering coefficient over the connectivity k for 43 organisms. For all organisms, this distribution approximates k^{-1}, which equals the distribution of artificial hierarchical networks. From these results, they conclude that metabolic networks are hierarchically organized. This hierarchy is achieved by modules (highly inter-connected regions), which are connected by hubs.

It is important to note that these properties do not describe metabolism itself, but the metabolic graphs that represent metabolism. Thus, they depend on the graph type that has been chosen to measure those properties. For example, the small-world property is highly dependent on the treatment of ubiquitous compounds. Ubiquitous compounds are the hubs of metabolic graphs. If they are avoided to obtain pathways between nodes of metabolic graphs that are closer to biochemical pathways, the small-world property vanishes. This is why two authors [Arita 04], [Croes et al. 06], recently contradicted the small-world property. After treatment of ubiquitous compounds, they obtained network diameters of 8 steps ([Arita 04], graph with reversible arcs) and around 7 steps ([Croes et al. 06], weighted KEGG graph). When Arita included ubiquitous compounds in his metabolic graph by ignoring the structural information of compounds, he measured a network diameter similar to the 3.2 steps observed by Jeong and co-workers, whereas Croes obtained a diameter of around 2 steps for his raw graph.

1.4 Related tools

There are two categories of tools available that perform tasks related to the task of the PathwayBuilder. It is therefore of interest to point out what they can do and how they differ from the PathwayBuilder.

1.4.1 Tools mapping microarray data on pathway maps

The first category includes tools that have been developed for the interpretation of microarray data with respect to metabolic pathways. A selection of them is listed in Table 1.2.

Tool	Web-tool	Stand alone	Micro-array data	Pathway data	Display	Remarks
GenMAPP (Gene Map Annotator and Pathway Profiler) [DAHLQUIST ET AL. 02]	no	yes	raw format	MAPPs (downloadable or user-defined pathways)	coloring of reactions according to expression values	only available for Windows
PathwayAssist [NIKITIN ET AL. 03]	no	yes	various formats	KEGG or user-defined pathways	coloring of reactions according to expression values	commercial software
MAPMAN [THIMM ET AL. 04]	no	yes	raw format	maps (downloadable or user-defined)	color-coded gene expression values for each condition	specialized on plant metabolism
Omics Viewer [PALEY & KARP 06]	yes	yes	raw format	BioCyc	highlights areas on organism-specific overview diagram	animated visualization possible

Table 1.2: Selection of tools, which map microarray data on metabolic pathways.

These tools share some common features. Usually, reaction nodes in the pathways are colored according to the expression value of the gene that codes for the enzyme, which carries out the given reaction. Many of them accept user-defined pathway data in addition to an inbuilt set of pathways [DAHLQUIST ET AL. 02], [NIKITIN ET AL. 03], [THIMM ET AL. 04]. All of them map gene expression data on pre-defined metabolic pathways (maps). Thus, they do not allow inference of metabolic pathways, which means that they are restricted to annotated pathways. If a group of co-expressed genes is associated to reactions located on different maps, these tools display at best separate maps but not a pathway that would connect these reactions. Here lies the improvement of the PathwayBuilder in comparison to these tools.

1.4.2 Tools for pathfinding in metabolic graphs

Tools in the second category infer pathways from metabolic graphs given a start and an end node. An incomplete list gives an overview on them.

Tool	Web-tool	Stand alone	Data	Algorithm	K shortest path	Search	Validation
BioMiner [SIRAVA ET AL. 02]	no	no	KEGG	depth first search with constraints	yes	2-end	2 pathways
PathMiner [MCSHAN ET AL. 03]	yes	no	KEGG	transition-space is searched by optimizing a score function	no	2-end	4 pathways

Tool	Web-tool	Stand alone	Data	Algo-rithm	K shortest path	Search	Validation
Metabolic Map Viewer [ARITA 03]	yes	no	KEGG, BRENDA, ENZYME	Eppstein	yes	2-end	coverage of reference metabolism (*E. coli*) and validity of paths checked (paths are regarded as valid if at least one carbon atom is transferred)
Pathway Hunter Tool [RAHMAN ET AL. 04]	yes	no	KEGG, BRENDA, PROSITE	breadth first search with constraints (chemical similarity)	yes	1-end and 2-end	2 pathways
Metabolic Pathfinding Tool [CROES ET AL. 05] [CROES ET AL. 06]	yes	no	KEGG, EcoCyc	backtracking with constraints	yes	2-end	on pathways stored in aMAZE (56) and EcoCyc (92)

Table 1.3: Selection of metabolic pathfinding tools.

Most tools listed in Table 1.3 are based on a k shortest path algorithm (Eppstein, backtracking) to rank a set of shortest paths between two seed nodes. They are all restricted to two seed nodes (2-end search) in contrast to the PathwayBuilder, which should accept any number of seed nodes. A major drawback of those tools is their lack of validation with respect to the inferred pathways, with the exception of the Metabolic Pathfinding Tool [CROES ET AL. 05]. Details on this tool will be given in the next section.

1.5 Previous projects

The current work depends on two other projects that have been carried out at the SCMBB.

1.5.1 aMAZE project

The aim of the aMAZE project was to integrate biological data from different sources in a generalized data model (introduced by [VAN HELDEN ET AL. 00], developed by [LEMER ET AL. 04a]) that allows complex queries. Those queries can be written in a newly developed query language called IQL (Igloo Query Language), which allows direct retrieval of graphs from the stored data [LEMER ET AL. 04b]. The aMAZE database, developed as part of the aMAZE project (http://www.scmbb.ulb.ac.be/amaze/), contained data imported from KEGG and in addition a number of metabolic pathways that were annotated based on the biological knowledge from human experts. The current work relies on the aMAZE project and related projects (transMAZE/bioMAZE) in several aspects: First, the metabolic graph was collected from the aMAZE database with the help of IQL. Second, the PathwayBuilder uses data structures developed by the Northbears team (http://www.northbears.org/) as part of the transMAZE/bioMAZE projects, which ease visualization and storage of graphs. Third, some of the annotated pathways stored in the aMAZE database have been used as references for the study cases. Details will be given in the following chapters.

1.5.2 Metabolic Pathfinding Tool

This work is based on previous work done by Didier Croes and Fabian Couche. Fabian Couche developed a variant of the backtracking algorithm that takes into account constraints like the path length and the maximal weight (more details in the material and methods chapter). Didier Croes' major contribution to metabolic pathway inference is the use of weighted graphs for the exclusion of ubiquitous compounds. For each of the data sets of metabolic reactions in KEGG and EcoCyc (the PGDB in BioCyc dedicated to *E. coli*), he constructed three different directed, bipartite graphs and compared the performance of the backtracking algorithm on those six graphs. The three graph types differed in their treatment of ubiquitous compounds. In the first graph (termed raw graph), ubiquitous compounds were not treated at all. In the second graph, called filtered graph, 36 ubiquitous compounds were excluded. In the third graph (called weighted graph), a weight was assigned to compound nodes that corresponds to their connectivity. The average positive predictive value and the average sensitivity of metabolic pathfinding were derived comparing inferred pathways with annotated pathways in aMAZE (56 pathways) and EcoCyc (92 pathways). The weighted graph had the highest average positive predictive value and sensitivity, both for its construction from the KEGG LIGAND database (validation with pathways in aMAZE) and from the EcoCyc database (validation with pathways in EcoCyc).

2 Material and Methods

2.1 Material

2.1.1 Metabolic database

The metabolic database of choice is the aMAZE database [Lemer et al. 04a], because of its advanced query language that allows the construction of bipartite, directed graphs directly from the whole set of reactions and compounds stored in aMAZE. The aMAZE database contains KEGG data imported in December 2004. In the near future, more recent updates will be used.

2.1.2 Language of implementation

Java (version 1.5) has been chosen as language for the implementation of the PathwayBuilder for several reasons:

Java is object oriented, which supports modular programming. To split a program in modules is of advantage if parts of the program should be altered or re-used. Java is portable, available for free and Java objects can be exchanged between multiple platforms.

More important, Java is the language of choice of the aMAZE and parts of the TransMAZE/BioMAZE projects, which includes data structures (especially the Graph, Data and GraphDataLinker structures) and libraries that are important for the current work. Using Java makes them directly accessible.

In addition, the backtracking algorithm developed by Fabian Couche, which forms the core of both the PathwayBuilder and the Metabolic Pathfinding Tool by Didier Croes, has been written in Java.

Thus, using Java eases the integration of previous work into the current project and code, which has already been developed and tested, does not need to be rewritten.

2.1.3 Integrated Development Environment

Eclipse (http://www.eclipse.org/) is an open source development platform that offers excellent support for Java. Among other services it provides content/code assist (autocompletion of variable names or common program structures like try/catch), quick fix (suggestions to correct errors), support for JUnit (for code testing), Ant (a make tool for Java) and refactoring (modification of code without changing the external behavior, for example change of a variable name). These features speed up development and make Eclipse the tool of choice for implementation in Java.

2.1.4 Graph visualization software

Cytoscape [SHANNON ET AL. 03], an open source graph visualization tool, is specialized in the visualization of biological networks. Its main advantage is its ability to link nodes and arcs with a number of attributes that are displayed separately in an attribute browser. It includes an implementation of several general graph layout algorithms, which allow to display graphs in a user-interpretable way. Cytoscape has been integrated into the aMAZE project by the North bridge, a plugin for Cytoscape that allows to send graphs coded in the GraphDataLinker structure to Cytoscape from a running program or to load graphs stored in GDL files into Cytoscape.

2.2 Methods

2.2.1 Linkage of genes to reactions

Given a group of co-expressed genes, the enzyme-coding genes need to be identified. If the function of the genes is known, any genome or pathway/genome database can be used to identify enzymes among the genes.
The more complicated step is then to link the enzyme to the reaction(s) it is catalyzing. One way to achieve this is to use the enzyme's EC number. The problem of linking enzymes to reactions via EC numbers is that one reaction can belong to more than one EC number and one EC number can be associated to more than one reaction. The latter is due to the fact that an EC number reflects a reaction mechanism rather than a concrete reaction. The same mechanism can take place with different co-factors or with different educts having the same functional group. For example the EC number 2.7.4.6 (ATP + nucleoside diphosphate $\Longleftrightarrow$ ADP + nucleoside triphosphate) is associated to 12 reactions, where the role of the nucleoside diphosphate is carried out by UDP, GDP, CDP and other nucleoside diphosphates.

A better way is the usage of annotated pathways where enzymes are linked directly to their reactions. For example in the aMAZE database, pathways have been annotated for three organisms (*E. coli*, yeast and human) that provide such a linkage, but they cover only a small part of known reactions and compounds for any of these organisms. Other annotation efforts attempt to reconstruct the whole metabolic network of an organism. For *S. cerevisiae*, Förster et al.[FÖRSTER ET AL. 03] provide a data set that links enzymes directly to reactions. It is based on information from genome annotation, pathway databases, biochemistry textbooks and recent publications. Unfortunately, the reactions are not cross-linked to KEGG reactions and therefore it is difficult to benefit from these annotations in the current work. However, this dataset might be useful for yeast-specific pathway-inference in the future.

Due to the lack of annotation, an enzyme-coding gene will be linked to its reaction(s) via its EC number(s). EC numbers reflect catalytic sites of enzymes. Hence, enzymes with more than one catalytic site are annotated with more than one EC number, each of which can be associated to more than one reaction.

There is support for the assumption that different catalytic activities of the same enzyme are usually involved in the same metabolic pathway (fusion enzymes, [Croes 05], chapter VI.2.2). Therefore, it is reasonable to take EC numbers as input for pathway inference. This requires a grouping of reactions that will be discussed in section 2.2.6.

2.2.2 Representation of metabolic data

As the work of Didier Croes demonstrates, the choice of a directed, bipartite and weighted graph for the representation of metabolic data allows successful metabolic pathfinding. None of the other graph structures described in the literature has been validated in a similar exhaustive manner with respect to metabolic pathfinding. Thus, the graph structure of Didier Croes was adopted for the current work. Below, a motivation for the choice of this particular graph structure is given. A more detailed description can be found in the thesis of Didier Croes [Croes 05].

Directed graph

In the context of pathway inference, a metabolic graph should be directed. Otherwise, a path going from educt to educt (or from product to product) of the same reaction is possible, which will lead in most cases to biochemical invalid paths. For example the reaction:
Cystathionine + Succinate $\Longleftrightarrow$ O-Succinyl-L-homoserine + L-Cysteine (R02508)
has two educts. An undirected graph would allow the following path:
Cystathionine $\Longleftrightarrow$ R02508 $\Longleftrightarrow$ Succinate
This path suggests that Succinate is synthesiszed from Cystathionine in one step, which is unlikely.

Bipartite graph

In a bipartite graph, compounds and reactions are represented as two separated node sets. In graphs with only one node set, either the compounds or the reactions need to be represented by the arcs. In case arcs are standing for reactions, each reaction occurs several times (as often as there are educts for the given reaction). The same holds if arcs stand for compounds: each compound occurs as often as there are reactions of which this compound is an educt. In metabolic graphs with only one node set it is therefore possible to cross either the same reaction or the same compound more than once. This complicates pathfinding unnecessarily.

Weighted graph

As has been shown in the introduction, different strategies have been applied to avoid ubiquitous compounds. In contrast to strategies based on chemical structure or rules, the use of weights does not require any additional information about the compounds and leads nevertheless to predictions of high accuracy. The connectivity has been chosen as weight for compound nodes by Didier Croes, because it is the only characteristic that differentiates ubiquitous from other compounds. Consequently, the k shortest path algorithm developed by Fabian Couche returns paths that are not shortest with respect to their length but with respect to their weight. Thus, the more connected a compound is (the more it behaves as an ubiquitous compound), the less likely it is to appear in a solution.

Reversibility of reactions

There are two strategies of dealing with directions of reactions: Either the direction can be represented as annotated in the metabolic database or for each reaction both directions can be included in the metabolic graph. The direction of a reaction depends on its free energy ΔG. ΔG in turn depends on the standard free energy ΔG_0 as well as on educt and product concentrations and the temperature ($\Delta G = \Delta G_0 + RT \ln \frac{[product_C]*[product_D]}{[educt_A]*[educt_B]}$). In principle, even a reaction with positive ΔG_0 is reversible if the term containing the ratio of product and educt concentrations is negative enough to outweigh ΔG_0. Only in a few cases ΔG_0, the educt and product concentrations and therefore the direction of reactions are known. Thus, the second strategy has been adopted for the collection of the metabolic graph. Reactions are represented as shown in Figure 2.1.

As can be seen in Figure 2.1, the graph is symmetric: The path A $\Longrightarrow$ R> $\Longrightarrow$ C has the same weight as the path C $\Longrightarrow$ R< $\Longrightarrow$ A. The exploitation of this symmetry saves computation time.

As pointed out in Didier Croes' thesis [CROES 05] and in [VAN HELDEN ET AL.02], the two directions of one reaction should be mutually exclusive (the choice of one of the two directions excludes the other from any solution). Otherwise, the same reaction could be passed twice. For example, the reaction:
Prephenate $\Longleftrightarrow$ Phenylpyruvate + H_2O + CO_2 (R01373)
can be considered. If its two directions are not treated as mutually exclusive, a path traversing the following nodes is possible:
Phenylpyruvate $\Longrightarrow$ R01373 reverse $\Longrightarrow$ Prephenate $\Longrightarrow$ R01373 direct $\Longrightarrow$ H_2O
Phenylpyruvate cannot be converted into H_2O in two reaction steps, therefore this solution is biochemical invalid.

A third strategy would be to use direction annotations as far as available and to rep-

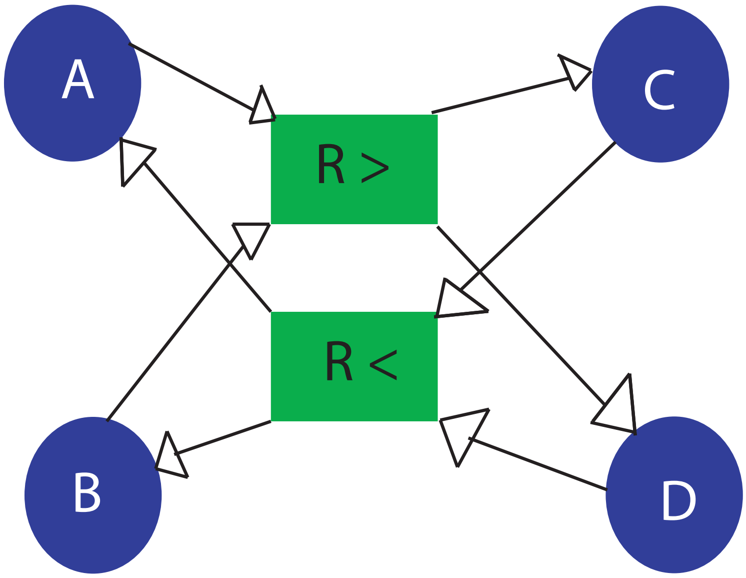


Figure 2.1: This Figure shows how direct and reverse reactions are represented in the metabolic graph.

resent reactions with unknown direction as described above. This strategy increases computation time because the symmetry of the metabolic graph will be destroyed. To answer the question whether the integration of more information outweighs the loss of computation time will be a task for the future.

2.2.3 Collection of the metabolic graph

The directed, bipartite and weighted metabolic graph containing direct and reverse reactions was collected from the aMAZE database using the Igloo Query Language (IQL). The metabolic graph consists of all compounds and reactions stored in aMAZE. To demonstrate the power of IQL, the query that was used to generate the metabolic graph used throughout this work is given below:

```
GET Reaction
HAVING Label ∼ ‘%‘
OR REVERSE_REACTION.Label ∼ ‘%‘
FILL ECNumber,ReferencedObject.PublicId,
< Educt.Compound,
> Product.Compound
INTO : aMAZE_metabolicGraph
```

2.2.4 K shortest path algorithm

The PathwayBuilder relies on the repetitive usage of a k shortest path algorithm. The k shortest path algorithm chosen is the constrained version of backtracking implemented by Fabian Couche.

Because the metabolic graph is large (22455 nodes) and backtracking is exponential, constraints are needed to reduce the size of the traversed graph.

In the implementation of Fabian Couche, the following constraints have to be satisfied by a valid solution path: The length of the path should not be larger than a maximum path length and not be smaller than a minimum path length. The weight of the path should not exceed a given maximum. The number of returned paths should be restricted by a given rank. In addition, a timeout constraint exists that interrupts the search after a given time to avoid too long running times in case there is no path between two seed nodes.

Given these constraints, the algorithm proceeds as follows ([COUCHE 02]):

- First, it tests whether the current node equals the end node. If this is the case, the path that has been followed from the start node to the current node is stored in a list of ranked paths.

- If the number of paths in the path list exceeds the number of ranks, the last (worst) path is removed from the list and the weight of the worst among the remaining paths is the new maximal weight.
- If the current node is not the end node, the algorithm tests whether the length of the path is below the maximal length. If this is the case, it tests for all non-marked neighbors of the current node, whether or not including this neighbor in the current path would increase its weight above the maximal weight.
- If not, the neighbor is marked as visited and added to the current path. The backtracking algorithm is then called on the neighbor as the new current node.

In contrast to pure backtracking, the constrained version has reasonable running times (in the order of minutes or limited by the time constraint).

2.2.5 Metabolic pathway inference algorithm

The task of the pathway inference algorithm is to connect a set of seed nodes in a metabolic graph. The solution is a subgraph of the metabolic graph that represents a metabolic pathway. This pathway could be branched or even contain cycles.

The pathway inference algorithm has to satisfy several requirements:

1. Firstly, it should accept multiple seed nodes. This contrasts with path finding, which is based on a pair of seed nodes (source and target).
2. It should be able to deal with an arbitrary order of seed nodes because a priori it is not known which seed nodes are the terminal nodes of the solution pathway and which are intermediate nodes.
3. It should be able to deal with orphan seed nodes. Orphan seed nodes are seed nodes that cannot be connected to any other seed node under the given constraints.
4. It should output a list of graphs, just as the k shortest path algorithm outputs a list of paths. Thus, alternative solutions can be explored.

These specific requirements make it hard to use a ready-made algorithm. However, Dooms and co-workers developed a promising approach to solve the problem of connecting multiple seeds in metabolic graphs [DOOMS ET AL.05]. CP(Graph), a constraint programming domain, allows formulating this problem in terms of constraints that are evaluated by dedicated propagators. Unfortunately, the current version of CP(Graph) is restricted to graphs with less than 500 nodes.

Van Helden and al. [VAN HELDEN ET AL.01] [VAN HELDEN ET AL.02] developed an algorithm that is able to connect a set of seed nodes in unweighted graphs. They applied this algorithm to a group of co-expressed genes from a gene expression data set in yeast [SPELLMAN ET AL. 98] and obtained a pathway that combined two known pathways, namely the sulfur assimilation and methionine biosynthesis pathways. Their algorithm was adapted to weighted graphs in the current work. Its principle is based on the idea by van Helden and colleagues, but details in the implementation might differ.

To explain the principle of the algorithm, the concept of a distance measure in graphs needs to be introduced. A distance measure is a nonnegative function that has three properties [CHARTRAND & LESNIAK 96]:

1. It is symmetric: $d(A,B) = d(B,A)$.
2. The distance between two objects A and B is zero iff A equals B.
3. The triangle inequality is valid: $d(A,C) \leq d(A,B) + d(B,C)$.

If a distance measure has been defined, a distance matrix consisting of the distances between all possible object pairs can be calculated. In the case of two seed nodes, the weight of the shortest path(s) connecting them has been chosen as distance measure.

The first step is to fill the entries in the distance and path matrices with the help of the k shortest path algorithm. Each entry of the distance matrix contains the weight of the shortest path(s) between two seed nodes. The path matrix stores all paths obtained for a given seed node pair under the given constraints.
This is the most time-consuming step, because for a metabolic graph with direct and reverse reactions, the k shortest path algorithm needs to be called $\frac{1}{2}N(N-1)$ times, where N is the number of seed nodes. This corresponds to one half of the matrix to be filled without diagonal. If the metabolic graph would contain only direct reactions or direct and reverse directions for only some reactions, even more entries of the matrix, namely $N(N-1)$ entries, would have to be calculated. This is the reason for the statement above that a metabolic graph containing direct and reverse directions for each reaction saves computation time.

In the next step, for each seed node its closest partner among all the seed nodes is identified. This information is contained in the distance matrix. From the path matrix, all paths between the seed node and its closest partner are collected. Paths that are redundant (which connect seed nodes already connected to other seed nodes by shorter paths) are removed.

The collected paths are assembled in the *guide graph*. Thus, the guide graph consists of all paths that connect a seed node to the closest among all other seed nodes. The guide graph can be filtered according to different criteria as the rank, the weight or the path length. The filtered guide graph is called *result graph* and represents the inferred pathway.

In addition to the retrieval of guide and result graph, a single linkage clustering is performed on the distance matrix. Single linkage [Sibson 73] is a bottom-up clustering technique that starts with clusters consisting of single objects (the leafs) and joins them until all objects to be clustered are contained in one single cluster (the root). The resulting tree is also called a dendrogram. The dendrogram shows, which seed node (or seed node group) is closest to which other seed node or seed node group.

In the following it is discussed whether this strategy can satisfy the requirements outlined above:

1. **Multiple seeds**
   The distance and path matrices can be calculated for N seed nodes, where N can be larger than two.

2. **Order of seed nodes**
   Another order of the seed nodes only changes the order of rows and columns in the distance matrix. The collection of node pairs with smallest distance from the distance matrix is independent from the order of its rows and columns.

3. **Treatment of orphan nodes**
   If a distance between two seed nodes cannot be obtained (timeout of the k shortest path algorithm), their distance is regarded as being infinite. A seed node with infinite distances to all other seed nodes is an orphan node.

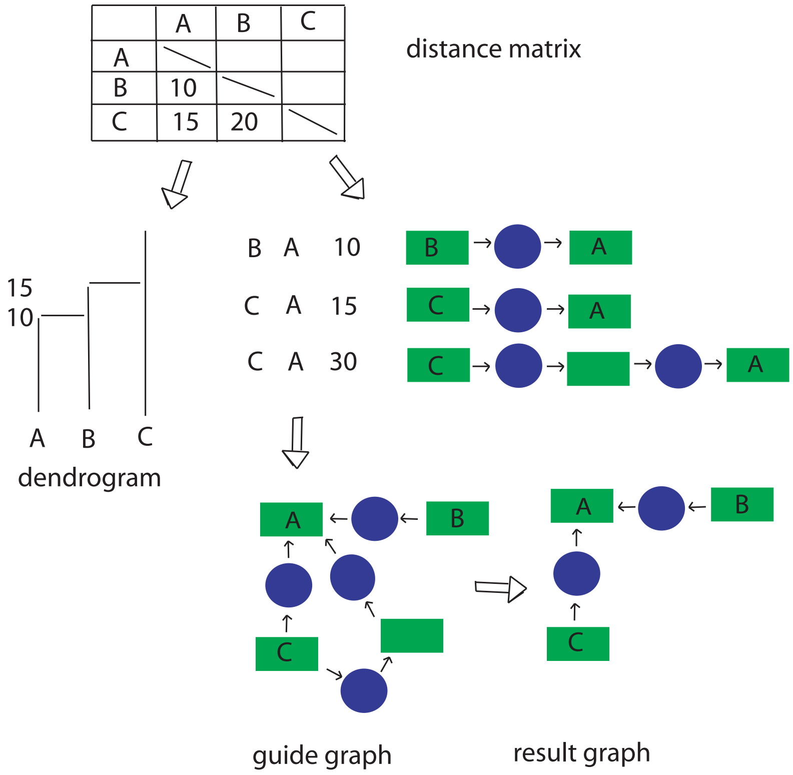
4. **Ranked list of subgraphs**
   The filtering of the guide graph allows obtaining a list, which contains the result graph of first rank, of second rank and so on.

The different steps of the algorithm are summarized in Figure 2.2.

2.2.6 Parameters of metabolic pathway inference

In addition to the parameters needed by the backtracking algorithm (rank, maximal weight, minimal and maximal length, time out) the following parameters are important:

- **Rank**
  It can happen that two seed nodes are connected by more than one path with



	A	B	C
A	\		
B	10	\	
C	15	20	\

Figure 2.2: The PathwayBuilder proceeds by first filling a distance and a path matrix for the given seed nodes. All paths that belong to the seed node pairs with shortest distance are unified to the guide graph. Paths of first rank form the result graph. In this example, one path of weight 10 was found between seeds A and B and two paths of weight 15 and 30 between seeds A and C. The distance of A and C is 15, because the distance between a seed node pair is defined as the weight of the shortest path connecting it. The shortest path between C and B with a weight of 40 is redundant, because B and C are already connected to other seeds (in this case A) by paths with weights below 40.

the same weight. These paths are equally valid solutions all of which should consequently be considered. This idea is expressed by the rank parameter: paths with the same weight have equal ranks. An example is given in the table below:

order (k) of paths	1	2	3	4
weight of paths	100	100	112	114
rank of paths	1	1	3	4

By default, the result graph is assembled from the guide graph by obtaining all paths of first rank. Note that paths could be ranked according to another parameter, for example their lengths. By default, the rank refers to the path weight.

- **Distance measure** The default distance measure between two seeds is the weight of the shortest path(s) connecting them. Alternative distance measures could be defined as for example the length of the path or a mixture of length and weight.

- **Node weights** By default, weights are given as described by Didier Croes, namely:

  - All the reaction nodes have a weight of 1.
  - The weight of a compound node corresponds to its degree.

  The weights could be modified, for example instead of the connectivity only the number of incoming or outgoing arcs could be taken as weight for compound nodes.

- **Path length** The path length can be defined in different ways. Usually, the length of a path is defined as the number of its nodes (including start and end node). An alternative definition of path length could take into account the number of arcs or the number of metabolic steps (reaction -> compound -> reaction or compound -> reaction -> compound).

2.2.7 Grouped seed nodes

As has been mentioned above, the PathwayBuilder needs to deal with grouped seed nodes. The following groups can occur in metabolic pathway inference:

1. **Reaction groups** Each reaction is a group consisting of its direct and reverse direction.

2. **EC number groups** In the context of pathway inference, an EC number is a group of reactions. Thus, an EC number group is a group of groups, because it consists of one or more reactions and each reaction is a group consisting of two directions.

3. **Enzyme groups** Enzymes can have multiple catalytic sites, in which case they are associated to more than one EC number. Consequently, an enzyme group contains all EC numbers associated to it. Enzyme groups can be separated into EC number groups, because multiple catalytic sites of one enzyme are likely to play a role in the same pathway. Thus, this group is not of concern for metabolic pathway inference from a set of co-expressed genes.

Figure 2.3 shows how grouped seed nodes are treated.

In case of single seed nodes the distance corresponds to the weight of the shortest path(s) between them. If the distance between groups of seed nodes should be calculated, a definition for the distance between seed node groups is needed.

In case of two reaction groups, the definition is straightforward: The shortest distance between two reaction groups corresponds to all shortest paths of same rank that connect a member of reaction group A with a member of reaction group B. For EC number groups, the shortest distance between two groups can be defined as all shortest paths of same rank that connect any reaction direction within EC group A to any reaction direction within EC group B.

2.2.8 Architecture of the PathwayBuilder

The PathwayBuilder consists of several components, which handle different tasks. Some components that were already available could be integrated thanks to the aMAZE team. This concerns the visualization as well as the metabolic graph collection and the k shortest path algorithm. In addition, the PathwayBuilder makes use of a graph structure developed by the aMAZE team and termed GraphDataLinker (GDL). It has several advantages over other graph structures: Thanks to the North bridge plugin GDL files can be loaded into Cytoscape or send to Cytoscape from a running program. This allows to automatically display the resulting graphs with various layout algorithms, and to explore it at different levels of resolution. The GDL stores attributes of nodes and arcs separately from the graph to speed up graph algorithms. These attributes can be added at need to the solution graph. The GDL can be saved as GDL file in XML format, which eases exchange and storage of graphs.

The architecture of the PathwayBuilder is summarized in Figure 2.4.

2.2.9 Comparison of annotated to inferred metabolic pathways

There are three ways to compare an inferred pathway to an annotated one.

1. The first way is to use a graph comparison algorithm that assigns a score depending on the degree of matching between the two graphs.

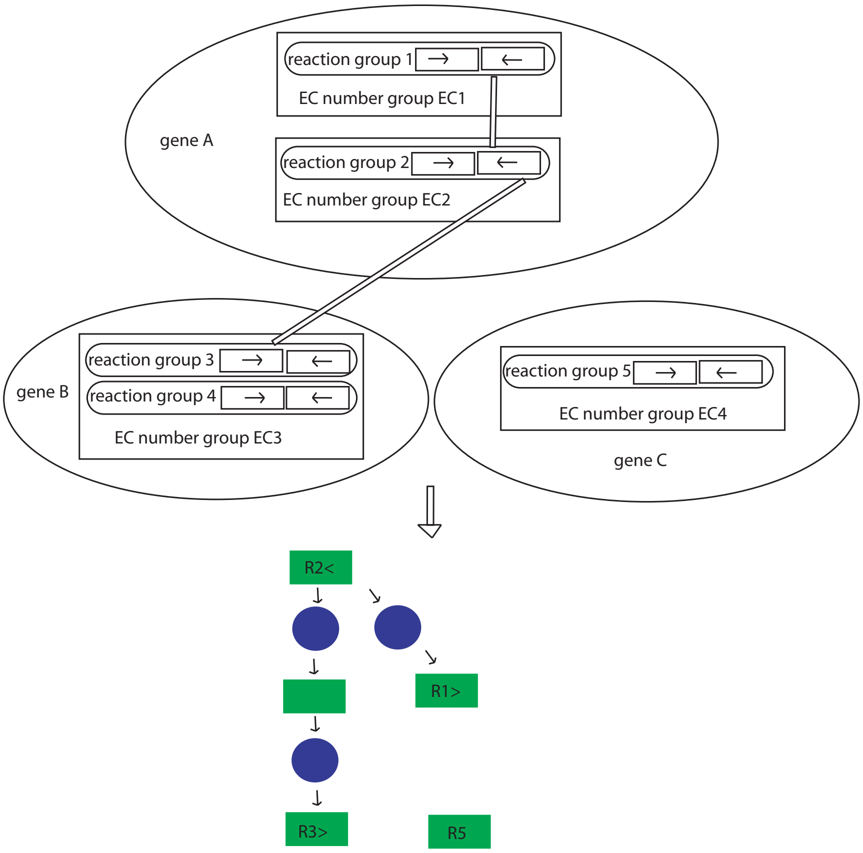


Figure 2.3: This Figure shows how a pathway is inferred from EC number groups. Three genes are given (A, B and C), which are linked to four EC number groups EC1, EC2, EC3 and EC4, which in their turn are associated to five reaction groups (R1 to R5). The PathwayBuilder finds shortest paths between reaction groups 1 and 2 and reaction groups 2 and 3, thereby linking EC number groups EC1, EC2 and EC3. Reaction group 5 cannot be linked to any other reaction group and is therefore an orphan. Below, the inferred pathway linking three of the four EC number groups is displayed. It is of note that EC number group EC3 contributes only one of its reactions to the inferred pathway.

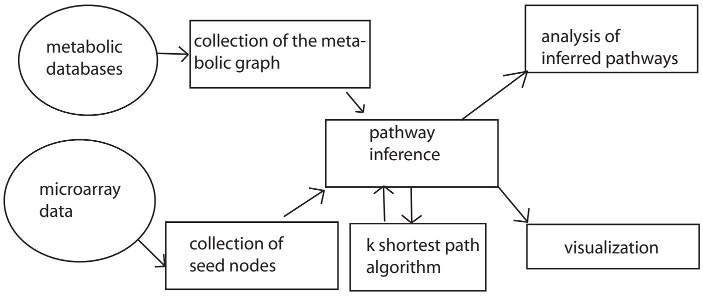


Figure 2.4: Architecture of the PathwayBuilder.

2. The second way is to linearize the annotated and inferred pathways and then to apply alignment algorithms on them. Both ways are beyond the scope of this final work, but might be explored in future.

3. The third and simplest way is to operate on the node sets of the annotated and the inferred pathway. This strategy does not take into account the order of nodes, but returns the number of true and false positives and false negatives. It has been used by Didier Croes to calculate the accuracy of his metabolic pathfinding tool. The size of the intersection of both sets gives the number of true positives (TP). The number of inferred nodes not present in the annotated pathway corresponds to the number of false positives (FP), and the number of annotated nodes not present in the inferred pathway to the number of false negatives (FN).

To evaluate a prediction tool, its sensitivity and specificity with respect to a reference data set needs to be calculated. The sensitivity indicates how likely the tool is to miss true positives. With increasing sensitivity, predictions will contain less false negatives. The specificity expresses how likely the tool is to accept false positives. The lower the specificity, the less likely the tool is to reject a false positive. An optimal tool should have specificity and sensitivity of 1.

Sensitivity

The sensitivity Sn is the ratio of true positives versus true positives and false negatives:
$Sn = TP/(TP+FN)$

Specificity

The specificity Sp is defined as the number of true negatives (TN) divided by the number of true negatives and false positives:
$Sp = TN/(TN+FP)$
In case of pathway inference, the number of true negatives corresponds to the number of all compounds and reactions (minus the compounds and reactions contained in the annotated and inferred pathway). Consequently, the ratio would always be close to 1, because the number of false positives is small in comparison to the number of true negatives. This is the reason why the specificity has been replaced by another measurement, namely the positive predictive value.

Positive predictive value

The positive predictive value PPV is defined as the number of true positives divided by the sum of true and false positives:
$PPV = TP/(TP+FP)$
Thus, it gives the ratio of true positives versus all positives.

Accuracy

Given specificity and positive predictive value, the accuracy Acc of the pathway inference can be calculated, which is defined as the mean of sensitivity and positive predictive value:
$Acc = (Sn+PPV)/2$

3 Results

In its current state, the PathwayBuilder accepts as input compounds, reactions and groups of reactions (EC numbers). Neither the linkage of genes to EC numbers nor the association of EC numbers to reactions is implemented in the moment, thus both steps need to be done independently with the help of pathway/genome databases. Given the input, the PathwayBuilder returns the inferred pathway in form of the result graph.

3.1 Properties of the metabolic graph

The metabolic graph on which the pathway inference is performed consists of 22455 nodes (11684 reaction and 10771 compound nodes) and 46430 arcs. It is of note that the number of reaction nodes and arcs has been doubled by the inclusion of direct and reverse reactions. Without the reverse reactions, the graph would include only 5842 reaction nodes and 23215 arcs.

A digraph D is strong (or strongly connected) if for every pair u, v of vertices, D contains both a $u-v$ path and a $v-u$ path. (cited from [CHARTRAND & LESNIAK 96]) The metabolic graph consists of 6534 strongly connected components (identified with the "Analyze Graph" function of the yED graph editor http://www.yworks.com/en/products_yed_about.htm). Thus, computation time can be decreased if at least components consisting of single compound or reaction nodes could be removed (which is planned as one of the future improvements).

The 10 compounds with highest connectivity in the given metabolic graph are listed below:

H_2O	3894
O_2	1346
H^+	1202
NAD^+	1190
NADH	1154
ATP	888
Orthophosphate	736
CO_2	716
ADP	652
CoA	624

This list differs from other lists of top 10 hubs as given in [Arita 04]. The list depends on the data from which the metabolic graph was constructed. A difference to previous lists might therefore be due to the different data sets used or to the different ways the graphs were constructed.
It is of note that there is a large distance between the most highly connected compound (H_2O) and the second most highly connected compound (O_2).

3.2 Study Cases

3.2.1 Pathway inference parameters

All the analyses were performed with the same parameters given below:

Maximal weight	1000
Maximal length	20
Minimal length	1
Number of ranks	5
Time-out	5 minutes

The maximal weight excludes a number of compounds (H_2O, O_2, H^+, NAD^+ and NADH) from any possible solution path. Hence, metabolic pathways in which those compounds constitute necessary steps cannot be inferred. The limit of 1000 was chosen to speed up computation and will be increased after planned improvements have been implemented.

For each input EC number, all reactions listed in KEGG have been taken into account. The result graph was retrieved from the guide graph by keeping all paths of first rank.

3.2.2 Methionine Biosynthesis in *E.coli*

This study case has been chosen to demonstrate the functionality of the PathwayBuilder. The methionine biosynthesis pathway linearized by Didier Croes has been selected as reference pathway (Figure 3.1), because Didier Croes showed in his supplementary material (http://www.scmbb.ulb.ac.be/Users/didier/pathfinding) that pathway inference given two seeds did not recover the complete pathway. Since two seed reactions are not sufficient to infer the complete pathway, three seed reactions (the start reaction R00480, the end reaction R00946 and an intermediate reaction, R01777) have been given as input to the pathway inference.

In Figure 3.2, the dendrogram of the pathway inference is given. It can be seen that the intermediate reaction R01777 has a shorter distance to the end reaction of the

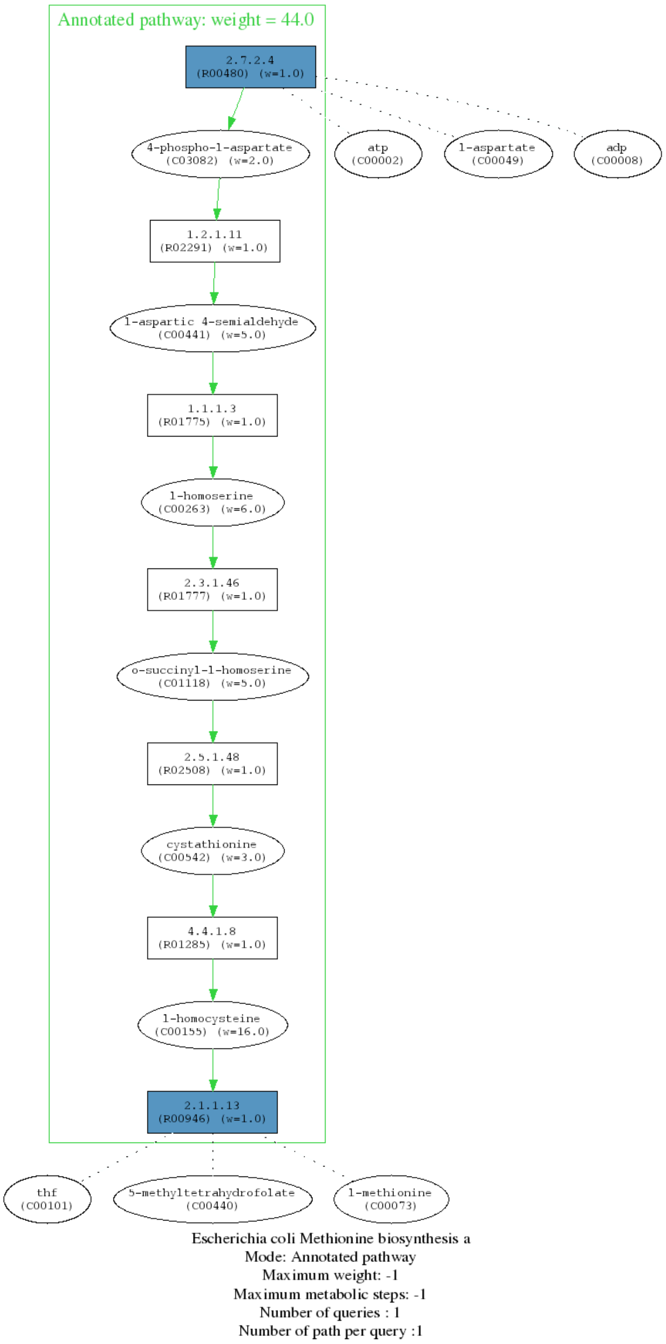


Figure 3.1: The annotated methionine synthesis pathway as given in the supplementary material of Didier Croes is shown.

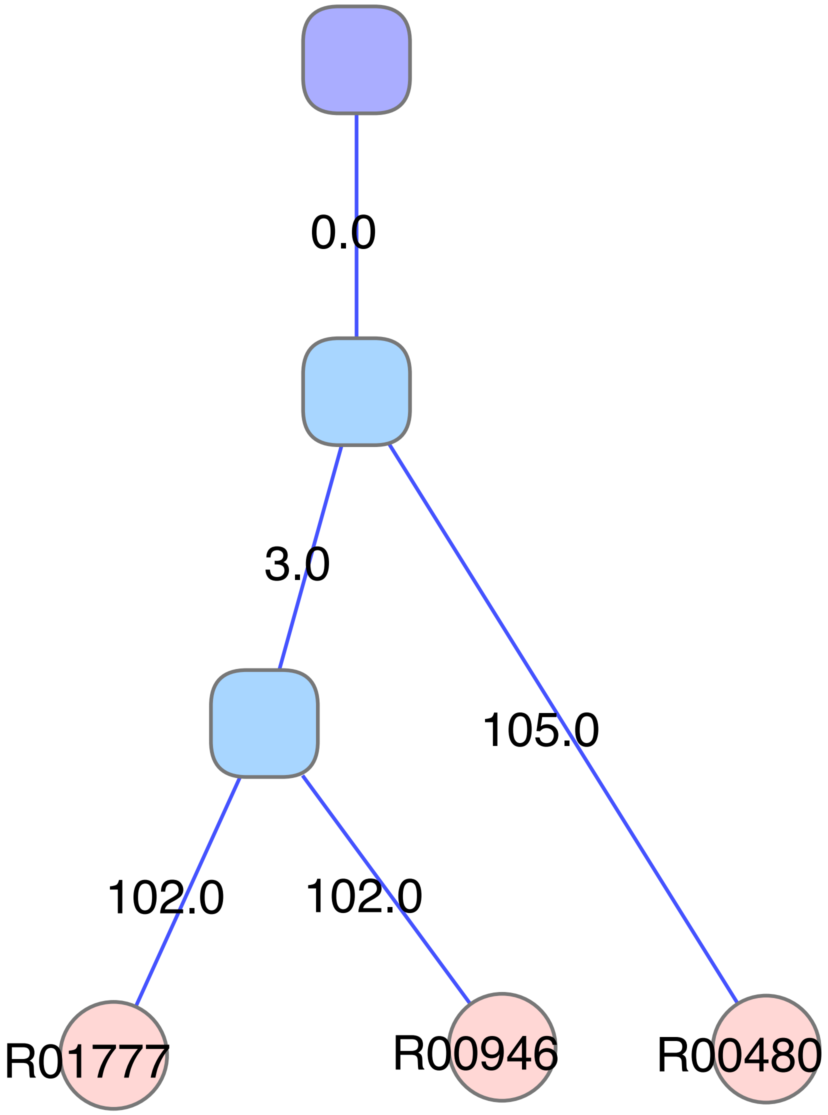


Figure 3.2: This figure shows the dendrogram of the methionine biosynthesis pathway inference. The edges of the tree representing the dendrogram are labeled with the distances at which two clusters are merged into one.

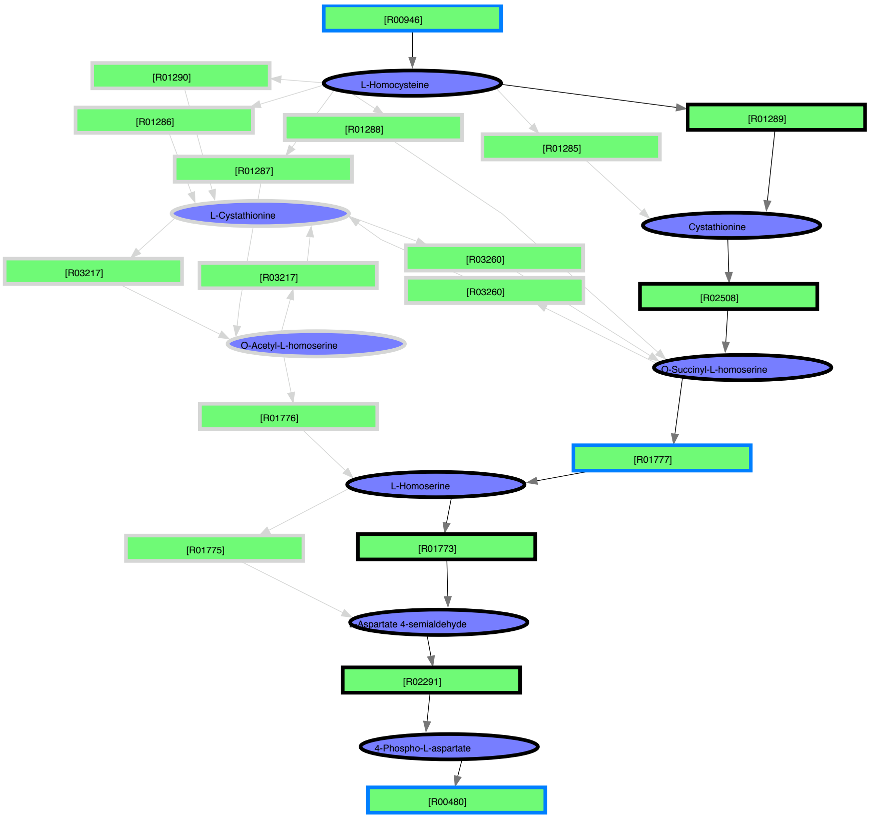


Figure 3.3: This figure shows the guide graph of the methionine biosynthesis pathway inference. Seed nodes have a blue, all other nodes a black border. Compounds are colored in blue and labeled with their name, reactions are filled with green color and labeled with their KEGG reaction id.

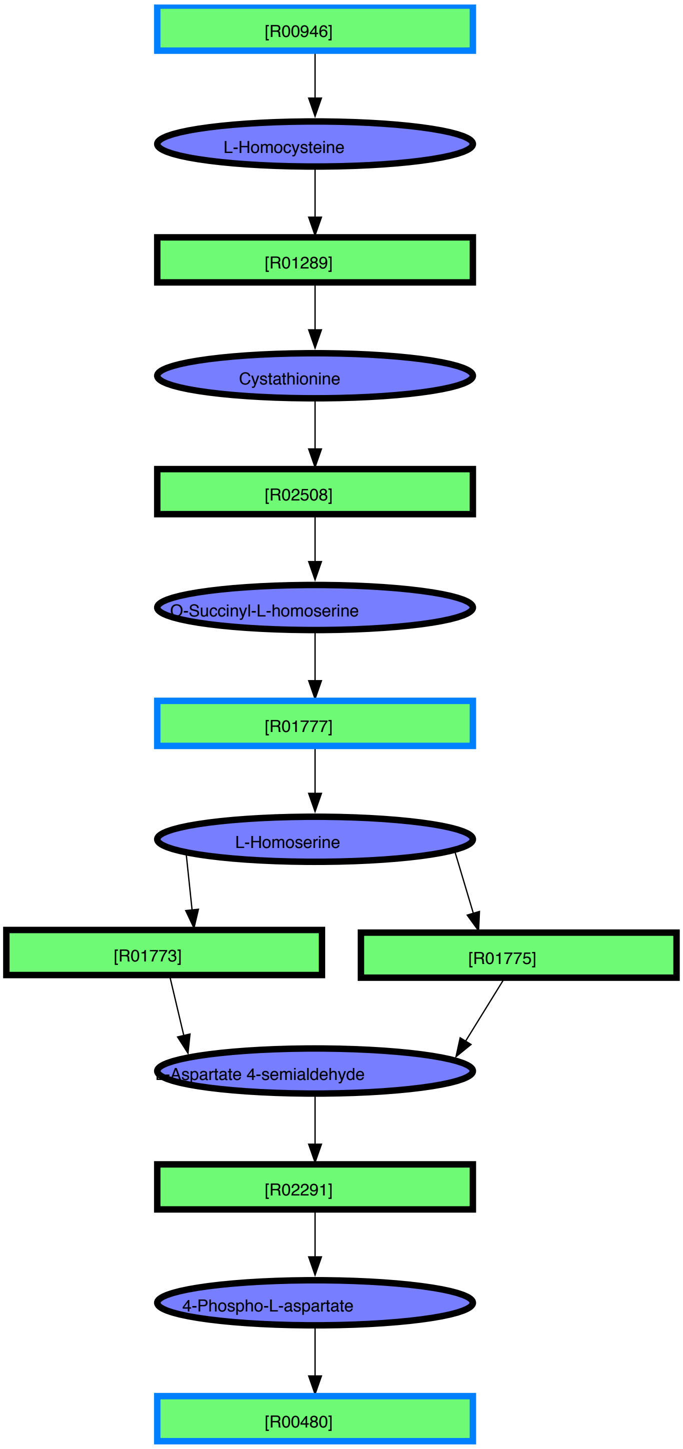


Figure 3.4: This figure shows the result graph of the methionine biosynthesis pathway inference. Seed nodes have a blue, all other nodes a black border. Compounds are colored in blue and labeled with their name, reactions are filled with green color and labeled with their KEGG reaction id.

annotated pathway than to the start reaction.

Figure 3.3 shows the guide graph of the pathway inference. In the guide graph, some alternative pathways can be seen that generate L-Homocysteine from 4-Phospho-L-aspartate via O-Acetyl-L-homoserine and L-Cystathionine.

The result graph is shown in Figure 3.4. Note that its direction differs from the annotated pathway. Since for each reaction its direct and reverse direction were included in the metabolic graph, the PathwayBuilder cannot differentiate between both directions and will return any of them arbitrarily.

A comparison of annotated and inferred pathway is given in Table 3.1. The inferred pathway goes from L-Aspartate 4-semialdehyde to L-Homoserine via two reactions (R01773 and R01775). Both reactions differ only by their co-factors (R01773 uses NAD^+ and R01775 *NADP*). Nevertheless, R01773 was counted as false positive, since it does not appear in the annotated pathway.

inferred pathway	annotated pathway
R00480	R00480
C03082	C03082
R02291	R02291
C00441	C00441
R01775	R01775
R01773	
C00263	C00263
R01777	R01777
C01118	C01118
R02508	R02508
C00542	C00542
R01289	R01285
C00155	C00155
R00946	R00946

Table 3.1: This table compares the inferred methionine biosynthesis pathway given three seed nodes to the annotated pathway. False positives are colored in red, true positives in green, false negatives in orange and seed nodes in blue.

When using two seeds, the inferred pathway of first rank given by Didier Croes skips Cystathionine (C00542) and the reaction that converts O-succinyl-l-homoserine (C01118) in Cystathionine (R02508). This gap could be filled by using three seed nodes.

The values for the sensitivity, the positive predictive value and the accuracy of the inferred pathway are given below:

Sensitivity	0.92
Positive predictive value	0.86
Accuracy	0.89

To compare the accuracy of pathway inference given three seeds to the accuracy given two seeds, the inference was repeated with two seeds. Surprisingly, the PathwayBuilder was able to infer the same pathway given two seeds only. Thus, the third seed node did not improve the inferred pathway and the higher accuracy in comparison to Didier Croes' result might be due to differences in the data set used.

3.2.3 Purine Biosynthesis in *E.coli*

The long purine biosynthesis pathway (containing 14 reactions) has been chosen as a more difficult study case. It demonstrates some of the problems pathway inference with multiple seeds is facing.

The purine biosynthesis pathway as annotated in BioCyc is shown in Figure 3.5. For this study case, the modified pathway given in the supplementary material of Didier Croes was taken as reference pathway (Figure 3.6). Didier Croes needed to linearize the branched pathway to be able to evaluate his two-end pathway inference tool. This linearization is not of relevance for the PathwayBuilder, since it accepts more than two seeds as input. However, this linearized pathway was chosen because Didier Croes demonstrated in his supplementary material that two-end pathway inference fails to reconstruct the whole pathway.

The following genes involved in purine biosynthesis are regulated by the repressor purR (information taken from BioCyc): guaA, guaB, purB, purC, purD, purE, purF, purH, purK, purL, purM and purN.

All or a subset of those genes might appear (negatively) co-expressed in a microarray experiment that measures the effect of adding purines to the medium on gene expression in *E. coli*. This is why input EC number groups for the pathway inferences described below were chosen from this gene set.

Purine Biosynthesis - pathway inference given 2 seeds

First, the two EC number groups containing the reactions annotated as start and end node were selected as input (associated reaction ids are given in brackets):
2.4.2.14 (R01072) and 4.3.2.2 (R01083, R04559)

Figure 3.7 shows the guide graph and Figure 3.8 the result graph of the pathway inference given these two EC number groups. The guide graph displays the paths from first to fifth rank, of which the first paths of first rank are highlighted in black. The small number of reactions in the result graph (six as compared to 14 in the annotated pathway) already shows that the pathway inference failed to recover the complete annotated pathway.

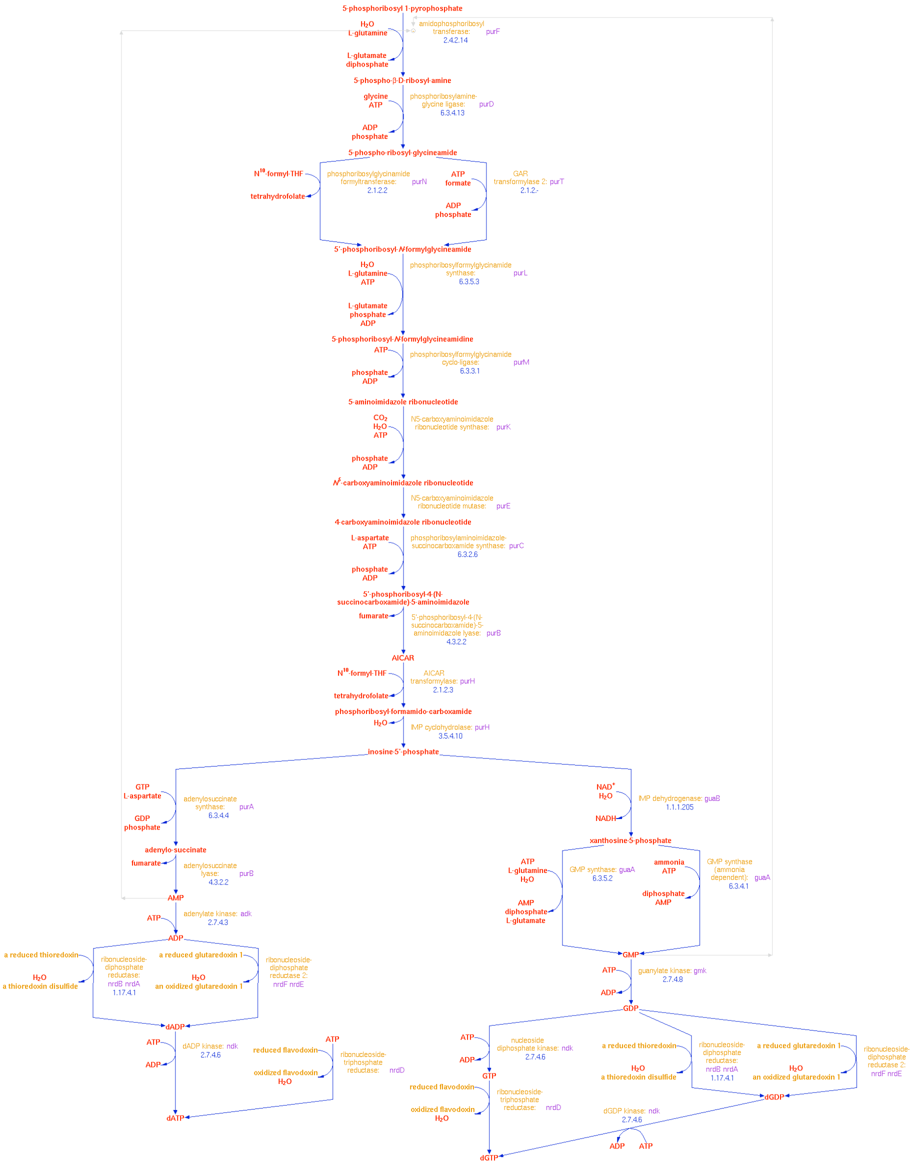

5-phosphoribosyl 1-pyrophosphate
H_2O
L-glutamine
L-glutamate
diphosphate
amidophosphoribosyl transferase: purF
2.4.2.14
5-phospho-β-D-ribosyl-amine
glycine
ATP
ADP
phosphate
phosphoribosylamine-glycine ligase: purD
6.3.4.13
5-phospho-ribosyl-glycineamide
N^{10}-formyl-THF
tetrahydrofolate
phosphoribosylglycinamide formyltransferase: purN
2.1.2.2
ATP
formate
ADP
phosphate
GAR transformylase 2: purT
2.1.2.-
5'-phosphoribosyl-N-formylglycineamide
H_2O
L-glutamine
ATP
L-glutamate
phosphate
ADP
phosphoribosylformylglycinamide synthase: purL
6.3.5.3
5-phosphoribosyl-N-formylglycineamidine
ATP
phosphate
ADP
phosphoribosylformylglycinamide cyclo-ligase: purM
6.3.3.1
5-aminoimidazole ribonucleotide
CO_2
H_2O
ATP
phosphate
ADP
N5-carboxyaminoimidazole ribonucleotide synthase: purK
N^5-carboxyaminoimidazole ribonucleotide
N5-carboxyaminoimidazole ribonucleotide mutase: purE
4-carboxyaminoimidazole ribonucleotide
L-aspartate
ATP
phosphate
ADP
phosphoribosylaminoimidazole-succinocarboxamide synthase: purC
6.3.2.6
5'-phosphoribosyl-4-(N-succinocarboxamide)-5-aminoimidazole
fumarate
5'-phosphoribosyl-4-(N-succinocarboxamide)-5-aminoimidazole lyase: purB
4.3.2.2
AICAR
N^{10}-formyl-THF
tetrahydrofolate
AICAR transformylase: purH
2.1.2.3
phosphoribosyl-formamido-carboxamide
H_2O
IMP cyclohydrolase: purH
3.5.4.10
inosine-5'-phosphate
GTP
L-aspartate
GDP
phosphate
adenylosuccinate synthase: purA
6.3.4.4
adenylo-succinate
fumarate
adenylosuccinate lyase: purB
4.3.2.2
AMP
ATP
adenylate kinase: adk
2.7.4.3
ADP
a reduced thioredoxin
H_2O
a thioredoxin disulfide
ribonucleoside-diphosphate reductase: nrdB nrdA
1.17.4.1
a reduced glutaredoxin 1
H_2O
an oxidized glutaredoxin 1
ribonucleoside-diphosphate reductase 2: nrdF nrdE
dADP
ATP
ADP
dADP kinase: ndk
2.7.4.6
ATP
reduced flavodoxin
oxidized flavodoxin
H_2O
ribonucleoside-triphosphate reductase: nrdD
dATP
NAD^+
H_2O
NADH
IMP dehydrogenase: guaB
1.1.1.205
xanthosine-5-phosphate
ATP
L-glutamine
H_2O
AMP
diphosphate
L-glutamate
GMP synthase: guaA
6.3.5.2
ammonia
ATP
diphosphate
AMP
GMP synthase (ammonia dependent): guaA
6.3.4.1
GMP
ATP
ADP
guanylate kinase: gmk
2.7.4.8
GDP
ATP
ADP
nucleoside diphosphate kinase: ndk
2.7.4.6
GTP
reduced flavodoxin
oxidized flavodoxin
H_2O
ribonucleoside-triphosphate reductase: nrdD
a reduced thioredoxin
H_2O
a thioredoxin disulfide
ribonucleoside-diphosphate reductase: nrdB nrdA
1.17.4.1
a reduced glutaredoxin 1
H_2O
an oxidized glutaredoxin 1
ribonucleoside-diphosphate reductase 2: nrdF nrdE
dGDP
dGDP kinase: ndk
2.7.4.6
ADP
ATP
dGTP

Figure 3.5: This figure shows the purine biosynthesis pathway in *E. coli* as annotated in BioCyc (Image taken from BioCyc).

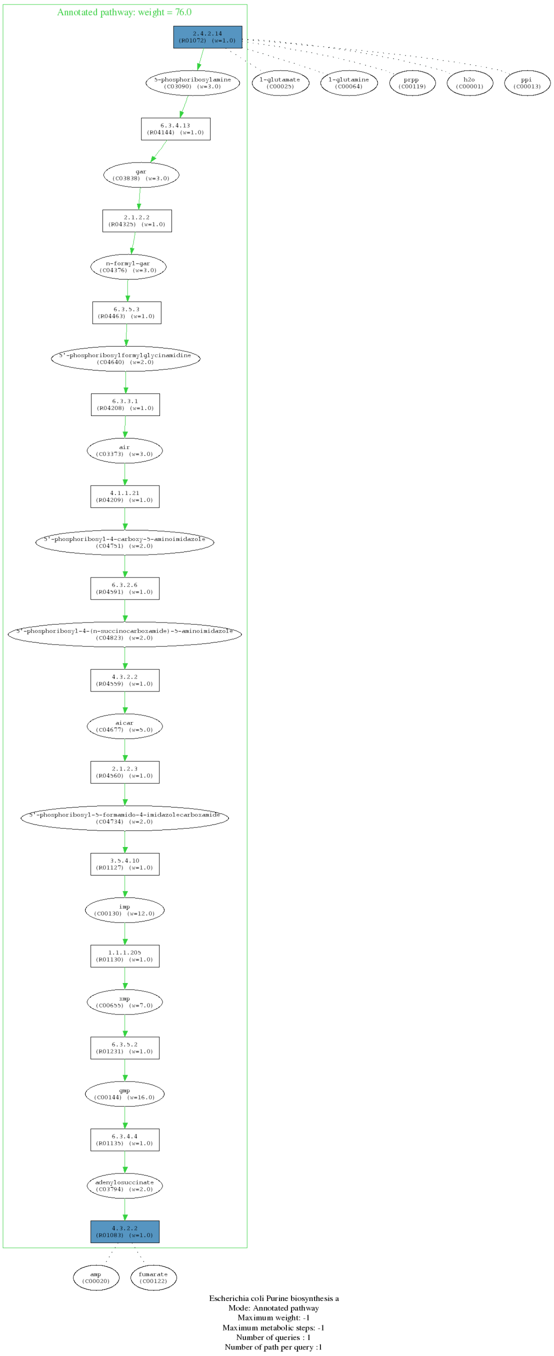


Figure 3.6: This figure shows the purine biosynthesis pathway in *E. coli* as annotated in the aMAZE database and linearized by Didier Croes. The figure has been taken from the supplementary materials of [CROES ET AL. 05].

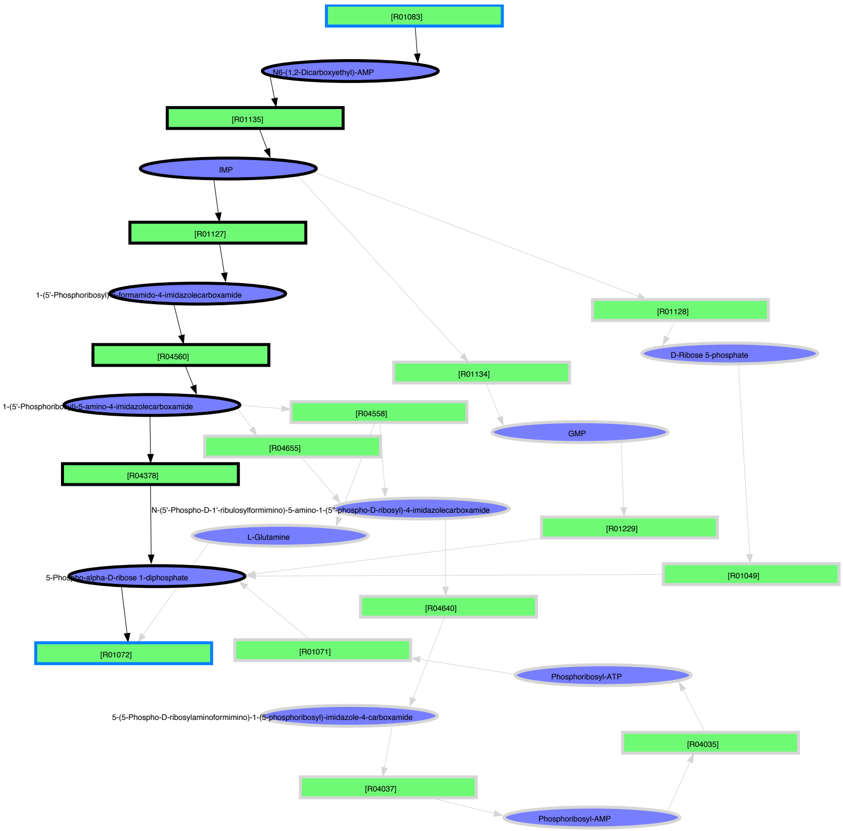

[R01083]
N6-(1,2-Dicarboxyethyl)-AMP
[R01135]
IMP
[R01127]
1-(5'-Phosphoribosyl)-5-formamido-4-imidazolecarboxamide
[R01128]
D-Ribose 5-phosphate
[R04560]
[R01134]
1-(5'-Phosphoribosyl)-5-amino-4-imidazolecarboxamide
[R04558]
GMP
[R04655]
[R04378]
N-(5'-Phospho-D-1'-ribulosylformimino)-5-amino-1-(5''-phospho-D-ribosyl)-4-imidazolecarboxamide
L-Glutamine
[R01229]
5-Phospho-alpha-D-ribose 1-diphosphate
[R01049]
[R04640]
[R01072]
[R01071]
Phosphoribosyl-ATP
5-(5-Phospho-D-ribosylaminoformimino)-1-(5-phosphoribosyl)-imidazole-4-carboxamide
[R04035]
[R04037]
Phosphoribosyl-AMP

Figure 3.7: The guide graph is shown, which contains all paths up to the fifth rank. The paths with k equals 1 are colored in black, the others in gray. Seed nodes have a blue border, compounds are colored in blue and reactions in green. The labels give the compound names and the reaction KEGG ids.

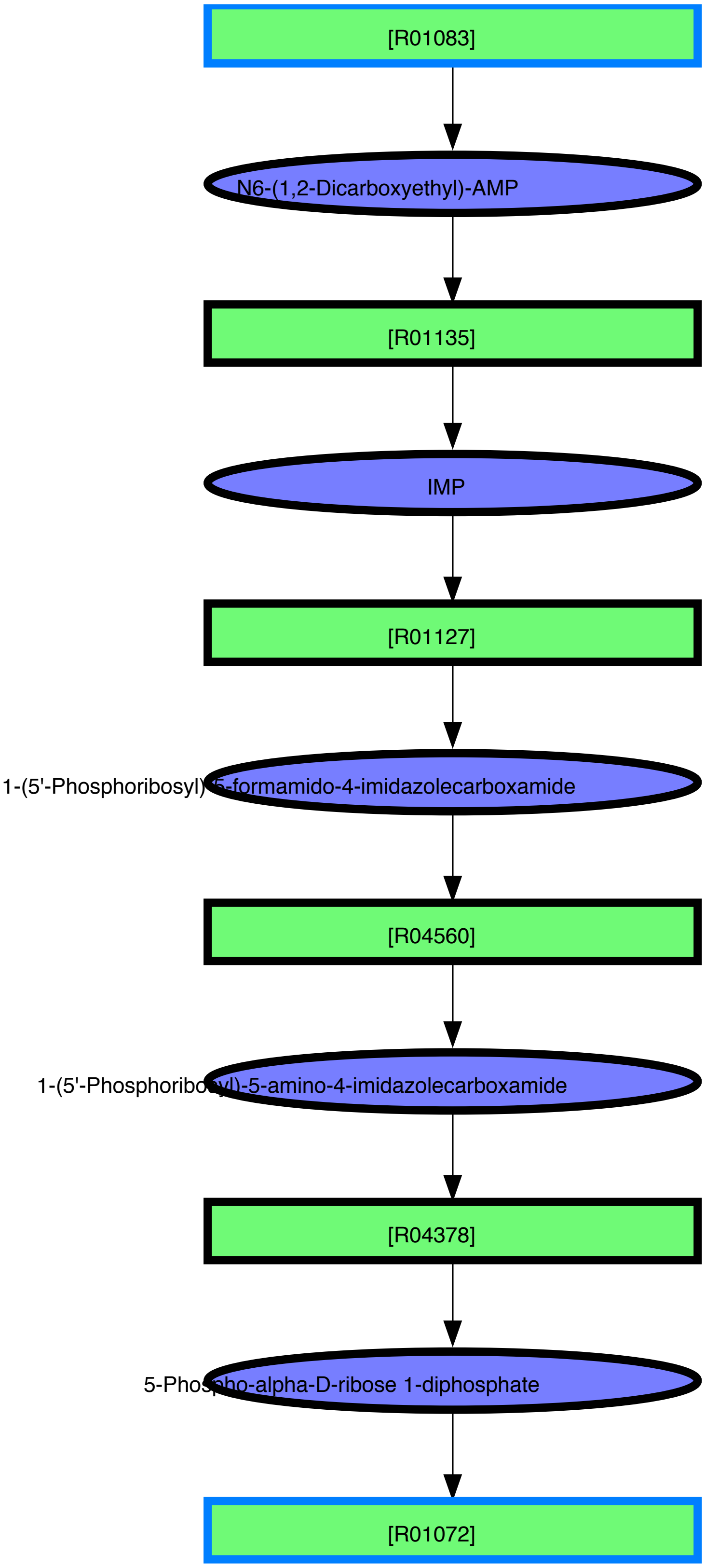

[R01083]
N6-(1,2-Dicarboxyethyl)-AMP
[R01135]
IMP
[R01127]
1-(5'-Phosphoribosyl)-5-formamido-4-imidazolecarboxamide
[R04560]
1-(5'-Phosphoribosyl)-5-amino-4-imidazolecarboxamide
[R04378]
5-Phospho-alpha-D-ribose 1-diphosphate
[R01072]

Figure 3.8: This figure shows the result graph that was retrieved from the guide graph by only keeping paths of first rank. Seed nodes have a blue border, compounds are colored in blue and reactions in green. The labels give the compound names and the reaction KEGG ids.

inferred pathway	annotated pathway
R01072	R01072
	C03090
	R04144
	C03838
	R04325
	C04376
	R04463
	C04640
	R04208
	C03373
	R04209
	C04751
	R04591
	C04823
	R04559
C04677	C04677
R04560	R04560
C04734	C04734
R01127	R01127
C00130	C00130
	R01130
	C00655
	R01231
	C00144
R01135	R01135
C03794	C03794
R01083	R01083
C00119	
R04378	

Table 3.2: This table compares the inferred purine biosynthesis pathway given two seed nodes to the annotated pathway. False positives are colored in red, true positives in green, false negatives in orange and seed nodes in blue.

Table 3.2 gives a comparison of the inferred versus the annotated pathway, highlighting false positives, false negatives and true positives by different colors. It illustrates that the inferred pathway is indeed far from reproducing the annotated pathway. This is also reflected by the sensitivity, the positive predictive value and the accuracy of the inferred pathway given below (values rounded):

Sensitivity	0.33
Positive predictive value	0.82
Accuracy	0.58

Interestingly, the PPV is quite high (0.82), indicating that most of the reactions and compounds in the annotated pathway are true positives. In contrast, the sensitivity is low, because two large parts of the annotated pathway are missing in the inferred pathway. One reason is the presence of two reactions associated to the same EC number in the annotated pathway. Both, R04559 and R01083, belong to EC number group 4.3.2.2. The PathwayBuilder, in its current implementation, allows only one reaction per EC number to contribute to the inferred pathway. Thus, a pathway containing two reactions associated to the same EC number group cannot be inferred correctly. However, additional seed nodes might help to reduce the size of the gaps.

Purine Biosynthesis - pathway inference given 3 seeds

Pathway inference was repeated with 3 EC number groups as input, given in the table below together with their associated genes and reactions:

gene	EC number	reactions
purB	4.3.2.2	R01083, R04559
purF	2.4.2.14	R01072
purM	6.3.3.1	R04208

The additional EC number (6.3.3.1) is associated to a reaction, which is located in a part of the annotated pathway that could not be inferred with two seeds only.

In the guide graph (3.9) two reaction nodes appear twice (R04325 and R04326). This occurs if both directions of the reaction take part in different paths and underlines the fact that directions of reactions cannot be inferred. It is of note that in the result graph (3.10) only three reaction nodes are marked as seed nodes. This illustrates that each EC number group can contribute only one of its reactions to the inferred pathway. In the case of EC number group 4.3.2.2, reaction R04559 rather than R01083 has been chosen by the pathway inference algorithm. This is also the reason why the lower part of the annotated pathway was missed (see Table 3.3).

The additional seed node increased the accuracy of pathway inference as the list below shows:

Sensitivity	0.44
Positive Predictive Value	0.86
Accuracy	0.65

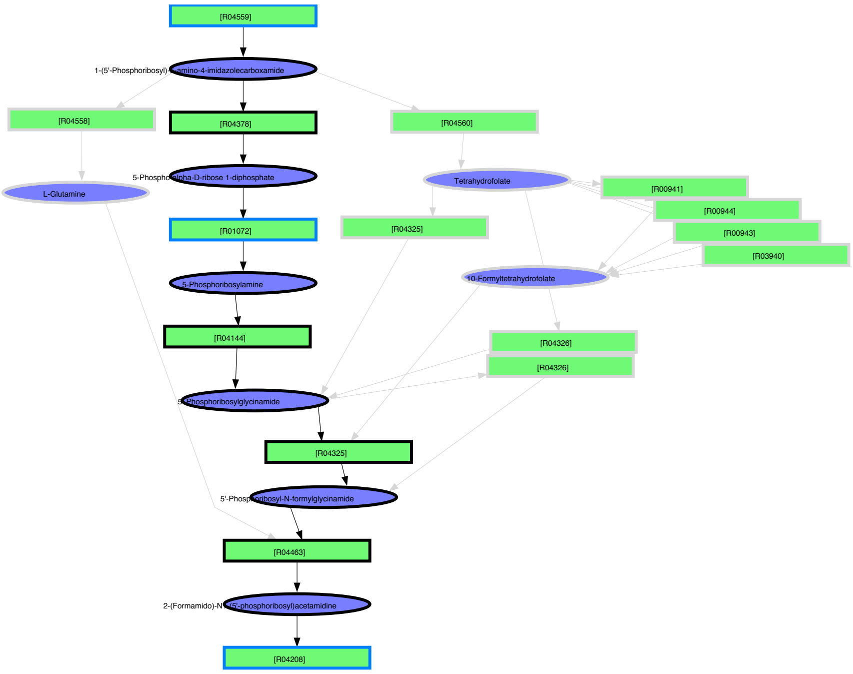

[R04559]
1-(5'-Phosphoribosyl)-5-amino-4-imidazolecarboxamide
[R04558]
[R04378]
[R04560]
5-Phospho-alpha-D-ribose 1-diphosphate
L-Glutamine
Tetrahydrofolate
[R00941]
[R00944]
[R00943]
[R03940]
[R01072]
[R04325]
5-Phosphoribosylamine
10-Formyltetrahydrofolate
[R04144]
[R04326]
[R04326]
5'-Phosphoribosylglycinamide
[R04325]
5'-Phosphoribosyl-N-formylglycinamide
[R04463]
2-(Formamido)-N1-(5'-phosphoribosyl)acetamidine
[R04208]

Figure 3.9: The guide graph for the pathway inference given 3 seeds is shown. Seed nodes have a blue border, compounds are colored in blue and reactions in green. The labels give the compound names and the reaction KEGG ids.

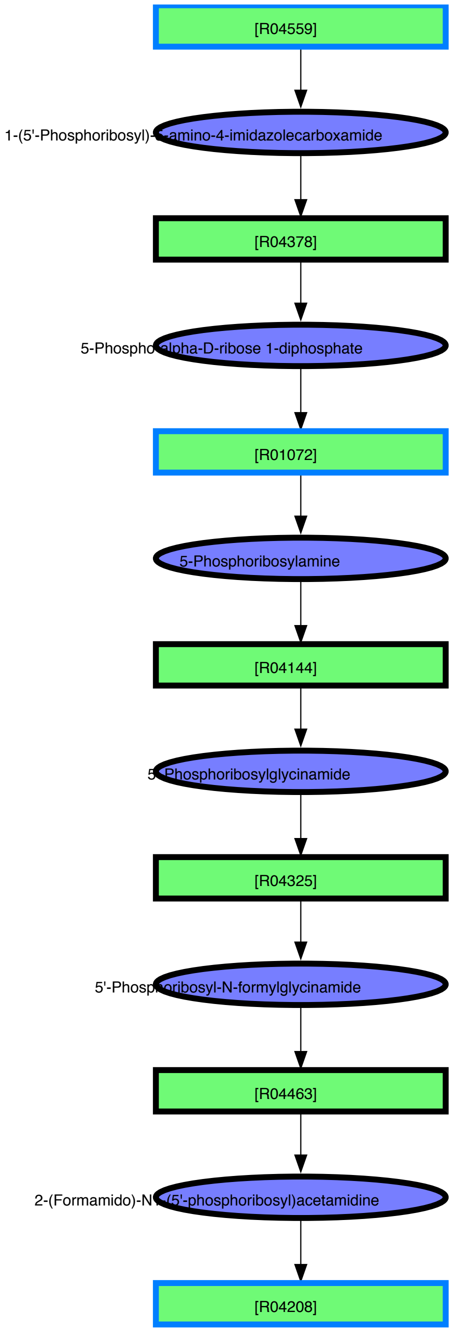

[R04559]
1-(5'-Phosphoribosyl)-5-amino-4-imidazolecarboxamide
[R04378]
5-Phospho-alpha-D-ribose 1-diphosphate
[R01072]
5-Phosphoribosylamine
[R04144]
5'-Phosphoribosylglycinamide
[R04325]
5'-Phosphoribosyl-N-formylglycinamide
[R04463]
2-(Formamido)-N1-(5'-phosphoribosyl)acetamidine
[R04208]

Figure 3.10: This figure shows the result graph of the pathway inference given 3 seeds. Seed nodes have a blue border, compounds are colored in blue and reactions in green. The labels give the compound names and the reaction KEGG ids.

inferred pathway	annotated pathway
R01072	R01072
C03090	C03090
R04144	R04144
C03838	C03838
R04325	R04325
C04376	C04376
R04463	R04463
C04640	C04640
R04208	R04208
	C03373
	R04209
	C04751
	R04591
	C04823
R04559	R04559
C04677	C04677
	R04560
	C04734
	R01127
	C00130
	R01130
	C00655
	R01231
	C00144
	R01135
	C03794
	R01083
C00119	
R04378	

Table 3.3: This table compares the inferred purine biosynthesis pathway given three seed nodes to the annotated pathway. False positives are colored in red, true positives in green, false negatives in orange and seed nodes in blue.

Purine Biosynthesis - pathway inference given 7 seeds

In the last test, the seven following EC number groups have been given as input to the pathway inference:

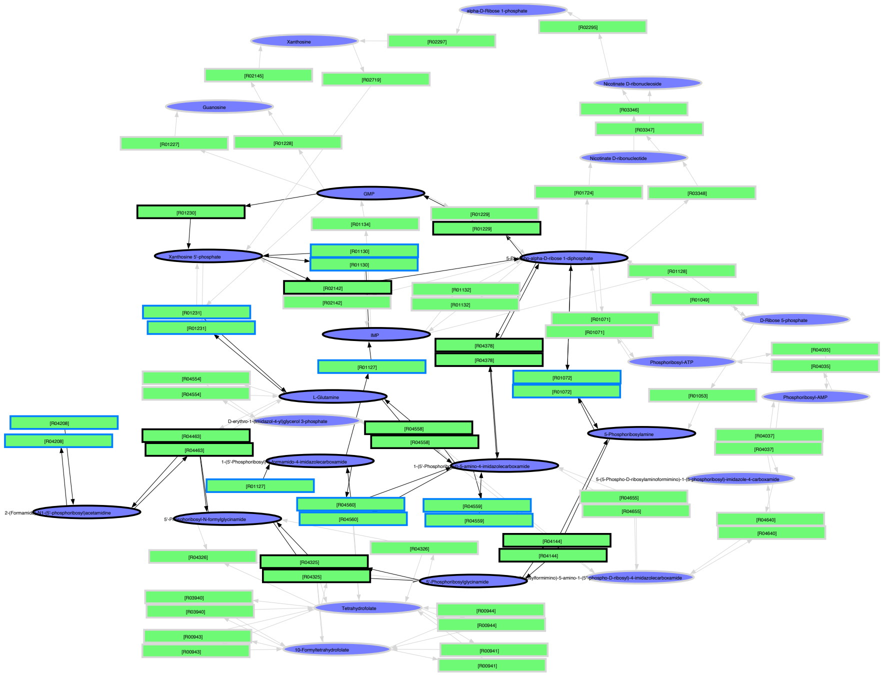


Figure 3.11: The guide graph for the pathway inference given 7 seeds is shown. Seed nodes have a blue border, compounds are colored in blue and reactions in green. The labels give the compound names and the reaction KEGG ids.

gene	EC number	reactions
guaA	6.3.5.2	R01231
guaB	1.1.1.205	R01130
purB	4.3.2.2	R01083, R04559
purF	2.4.2.14	R01072
purH	3.5.4.10	R01127
purH	2.1.2.3	R04560
purM	6.3.3.1	R04208

These seven genes form a group that could be found co-expressed in a microarray experiment. PurH demonstrates that an enzyme-coding gene can be associated to more than one EC number.

The resulting pathway (Figure 3.12) is more complex than any of the previous inferred pathways. Not only does it contain cycles, but some reaction nodes occur in both directions. It demonstrates that a more sophisticated approach of comparison with annotated pathways is needed, since it is not obvious how close the inferred pathway is to the annotated pathway.

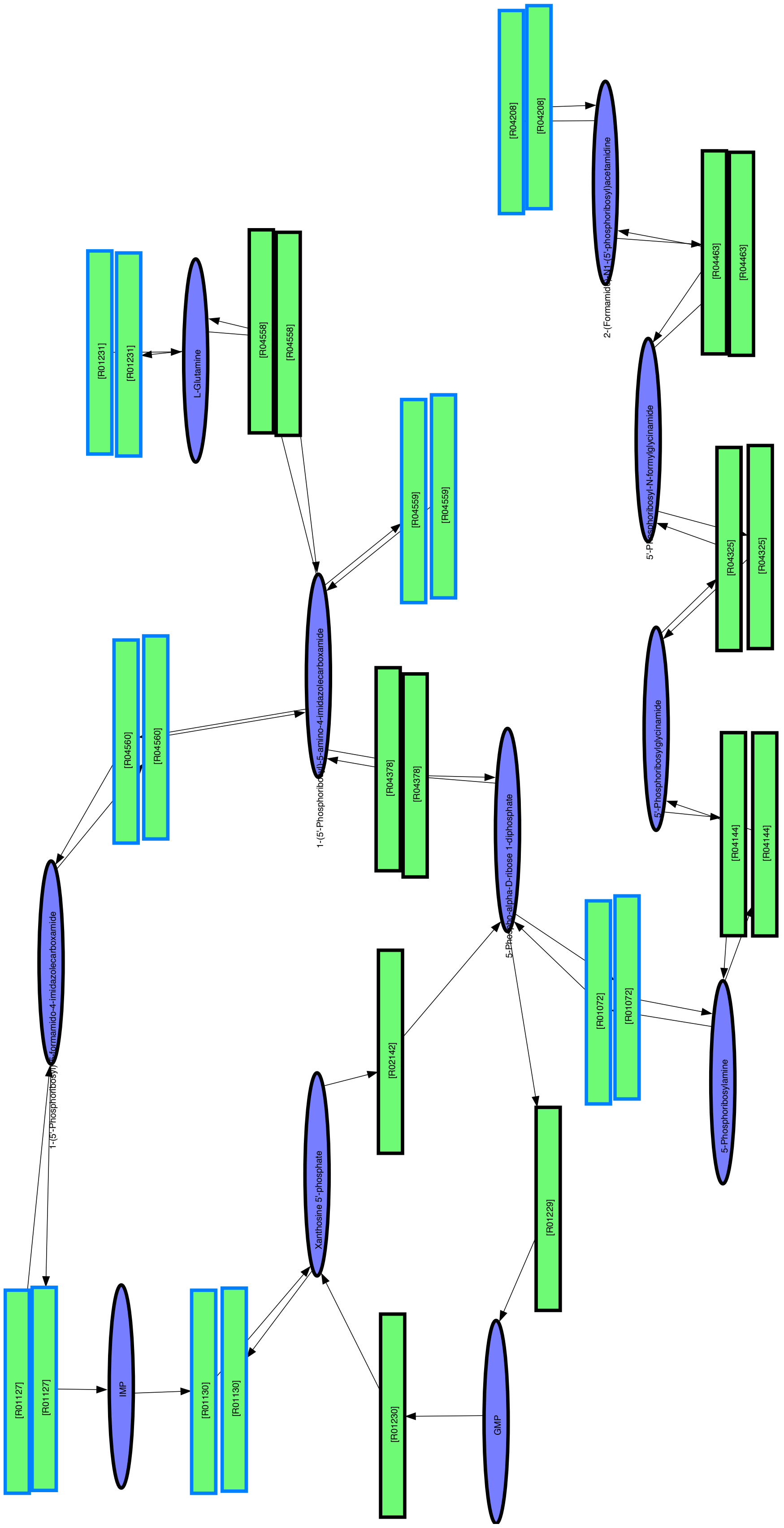


Figure 3.12: This figure shows the result graph of the pathway inference given 7 seeds. Seed nodes have a blue border, compounds are colored in blue and reactions in green. The labels give the compound names and the reaction KEGG ids.

inferred pathway	annotated pathway
R01072	R01072
C03090	C03090
R04144	R04144
C03838	C03838
R04325	R04325
C04376	C04376
R04463	R04463
C04640	C04640
R04208	R04208
	C03373
	R04209
	C04751
	R04591
	C04823
R04559	R04559
C04677	C04677
R04560	R04560
C04734	C04734
R01127	R01127
C00130	C00130
R01130	R01130
C00655	C00655
R01231	R01231
C00144	C00144
	R01135
	C03794
	R01083
C00119	
R04378	
R04558	
C00064	
R01229	
R02142	
R01230	

Table 3.4: This table compares the inferred purine biosynthesis pathway given seven seed nodes to the annotated pathway. False positives are colored in red, true positives in green, false negatives in orange and seed nodes in blue.

The comparison in Table 3.4 illustrates that the inferred pathway is still erroneous, but reproduces larger parts of the annotated pathway than any of the previously inferred pathways. The false positives R04378 and C00119 present in the previously inferred pathways occur again. They provide a shortcut from R04559 via 1-(5'-Phosphoribosyl)-5-amino-4-imidazolecarboxamide to R01072, thus avoiding reactions R04209 and R04591, which would have connected R04559 with R04208. The other gap (R01135 to R01083) is due to the problem described above, namely the presence of two reactions in the annotated pathway belonging to the same EC number group.

The accuracy is the highest achieved so far:

Sensitivity	0.7
Positive predictive value	0.73
Accuracy	0.71

The example of the purine biosynthesis pathway shows that additional seed nodes can improve the accuracy of the inferred pathway (in this study case from 0.58 given two seeds to 0.71 given seven seeds). It also demonstrates that decisions taken during the implementation of the PathwayBuilder prevent in some cases the correct inference of an annotated pathway.

4 Discussion

In this chapter, some aspects of pathway inference are treated in more detail that demand discussion (section 4.1 and 4.2) or that show its limits (section 4.3). The chapter ends with an overview on the next steps planned.

4.1 Advantages and disadvantages of pathway inference compared to pathway mapping

In contrast to the tools that map a set of co-expressed enzyme-coding genes on metabolic pathways, the PathwayBuilder attempts to infer the graph that best connects those enzymes in the metabolic network according to given criteria.

An advantage of this approach over simple mapping is that it overcomes the problem of the (often artificial) pathway boundaries. Co-expressed enzyme-coding genes do not necessarily respect these boundaries and might be located in pathways that are traditionally separated. The PathwayBuilder deals better with these situations than tools based on pathway maps.

In addition, it can find new combinations of known compounds and reactions that might result in variants of known pathways or in unknown pathways. These variants or new pathways might occur in organisms whose metabolism has not been described completely, or in mutants. Mapping tools cannot deal with variations.

The major disadvantage of pathway inference is that in contrast to mapping it can introduce errors, namely pathways that are biochemical invalid. This is the price, which is paid for the inference (prediction) of possibly new pathways. However, a large number of seed nodes might reduce the number of errors to a reasonable amount.

4.2 Open questions

Biochemical valid pathways

In this final work, the expression biochemical valid has been used to describe a correctly inferred pathway. But how can biochemical validity of a pathway be defined? Definitions like the one given by Arita (in a valid metabolic pathway at least one carbon atom should be transferred from the start compound to the end compound) [ARITA 04] are not helpful in case of branched pathways or cycles. In general, it could be stated that biochemical valid pathways are pathways that have been described in textbooks or databases based on observations. To know whether an unknown, inferred pathway is biochemical valid means to check whether a set of rules derived from valid pathways is satisfied by the new pathway. Rule-based pathway inference methods [MCSHAN ET AL. 03], [HOU ET AL. 04] return only those pathways that fulfill the given rules. But their set of rules does not need to be complete and might exclude pathways that occur in nature from their solution set. Pathway inference based on a metabolic graph implies rules as well. These rules are less restrictive than those imposed in [MCSHAN ET AL. 03] and [HOU ET AL. 04]. But if they are sufficient to infer known paths correctly, they might be as well sufficient to predict biochemical valid unknown paths. To give a detailed overview on definitions and rules introduced to describe biochemical valid pathways is out of scope of this final work. At least, this short discussion shows the importance of the notion of biochemical validity for pathway inference.

Collection of the metabolic graph

Another open question is whether organism-specific metabolic graphs or a graph including all known compounds and reactions should be used. In principle, this is a question of specificity versus sensitivity of the PathwayBuilder. Pathways inferred from an organism-specific metabolic graph are more likely to be correct than pathways inferred from a generic graph (high specificity). But there might also be more often cases in which no solution for the given set of enzymes can be found, because the organism-specific metabolic network might be incomplete (low sensitivity). A good trade-off between specificity and sensitivity might be achieved by using a metabolic graph collected from a set of related organisms.

How to derive the inferred pathway from the guide graph

The solution of the pathway inference is the result graph, which represents the inferred pathway. As has been explained in the chapter methods and materials, the result graph is obtained from the guide graph by only keeping the paths of best rank. The subset of paths, which is treated as solution, can be changed. For example, not only the paths of first rank, but of first and second rank or of second rank only could be regarded as solution. Didier Croes listed as result of his pathfinding tool the five paths of first to fifth rank and chose the best path among them as solution. Having additional information in form of more seed nodes allows defining the solution more strictly as all paths of first rank only. If this definition of a solution is optimal remains to be explored.

4.3 Limitations and problems

This section lists some generic limitations and some limitations due to certain decisions taken during implementation of the PathwayBuilder. In addition, unsolved problems are described.

Collection of seeds from microarray data

The set of co-expressed genes obtained from microarraymicroarraymicroarray data only contains those enzyme-coding genes that are regulated. Many reactions of a pathway might not be regulated, which leads to the gaps that the pathway inference algorithm attempts to fill. Some of these gaps might appear at crucial points in the pathway (where an additional seed node would have helped to prefer the correct over a lighter pathway). Thus, there might be cases where correct inference is not possible due to lack of seeds.

Directions of reactions

Since the collected metabolic graph contains both directions for each reaction, direct and reverse direction of the inferred pathway are equivalent solutions. Thus, an inferred pathway does not give any information about its direction.

Best solution

The pathway inference algorithm described is a heuristic to solve the problem of connecting a set of seed nodes in a graph. It does not necessarily find the best connection possible for the given criteria.

Treatment of EC number groups

In the current implementation of the PathwayBuilder, an EC number group can only contribute one of its associated reactions to the pathway to be inferred. This prevents the correct inference of pathways that contain two (or more) reactions associated to the same EC number. An alternative strategy would be to allow EC number groups to contribute more than one reaction to the pathway given certain conditions. For example, from two reactions A and B in EC number group 1, A could be closest to reaction C in EC number group 2 and B could be closest to reaction D in EC number group 3. In this case, A and B can both be part of the pathway and the PathwayBuilder would attempt to connect them. Whether or not this strategy would improve pathway inference needs to be explored.

Constraints

Constraints of the k shortest path algorithm (like maximal weight or maximal path length) reduce the solution space but at the same time they exclude solutions that might be biochemical valid. This could be improved by using a k shortest path algorithm that does not require constraints to achieve reasonable running times on graphs of the given size (for example Eppstein).

Speed of computation

The disadvantage of the current pathway inference algorithm is its repetitive call to an exponential algorithm, the backtracking. If backtracking could be replaced through a quicker k shortest path algorithm, this would speed up computation. Distributed computing using a cluster of computers could increase the speed as well. The computation of the k shortest paths between a seed node pair can be done in parallel for all possible pairs of seed nodes on different machines.

4.4 Outlook

Future improvements of the PathwayBuilder

The development of the PathwayBuilder is not yet finished. A module is needed that links input genes to reactions and another module that allows collection of metabolic networks from different databases. In addition, filtering of the metabolic graph should be possible to exclude certain reactions or compounds or to obtain a metabolic graph consisting of a set of organism-specific metabolic graphs. Cluster computing is under way but not yet finished, and a number of other improvements still needs to be done.

Parameter optimization

As has been described in the methods chapter, the PathwayBuilder depends on a number of parameters. For the parameters of the k shortest path algorithm, values have been recommended by Didier Croes. Other parameters as the distance measure, the node weights (i.e. zero weight for the seeds, additional weight for all other nodes) or the set of organism-specific metabolic graphs need to be optimized.

Validation

The next step will be the validation of the PathwayBuilder. The validation will quantify how close inferred pathways are to (annotated) reference pathways. The two databases presented in the introduction, KEGG and BioCyc will serve as source for the metabolic graphs and annotated pathways. It will be of interest to quantify how much (if at all) on average additional seed nodes improve the accuracy of the pathway inference. For the individual pathways, improvement of accuracy will depend on the position of additional seed nodes.

To take the validation one step further would be to apply the PathwayBuilder to microarray data. To evaluate its ability to infer metabolic pathways from sets of co-expressed genes, positive and negative controls are required. The pathways known to be up or down regulated under a given condition should be recovered from the positive control, whereas no biochemical valid pathway should be obtained from the negative control. Suitable data sets to achieve this remain to be identified.

Bibliography

[ARITA 03] M. Arita, *In Silico Atomic Tracing by Substrate-Product Relationships in Escherichia coli Intermediary Metabolism*, Genome Research **13** (2003), 2455–2466. 14, 20

[ARITA 04] M. Arita, *The metabolic world of Escherichia coli is not small*, PNAS **101** (2004), 1543–1547. 14, 16, 17, 40, 62

[CHARTRAND & LESNIAK 96] G. Chartrand and L. Lesniak, *Graphs & Digraphs*, third edition ed., Chapman & Hall, 1996. 4, 30, 39

[COUCHE 02] F. Couche, *Recherche de chemins sur les voies métaboliques*, Tech. report, Université Libre de Bruxelles, 2002. 28

[CROES 05] D. Croes, *Recherches de chemins dans le réseau métabolique et mesure de la distance métabolique entre enzymes*, Ph.D. thesis, Université Libre de Bruxelles, 2005. 25, 26

[CROES ET AL.05] D. Croes, F. Couche, S. Wodak, and J. van Helden, *Metabolic PathFinding: inferring relevant pathways in biochemical networks*, Nucleic Acids Research **33** (2005), W326–W330. 3, 15, 16, 20, 48

[CROES ET AL.06] D. Croes, F. Couche, S. Wodak, and J. van Helden, *Inferring Meaningful Pathways in Weighted Metabolic Networks*, J. Mol. Biol. **356** (2006), 222–236. 15, 16, 17, 20

[DAHLQUIST ET AL.02] K.D. Dahlquist, N. Salomonis, K. Vranizan, S.C. Lawlor, and B.R. Conklin, *GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways*, nature genetics **31** (2002), 19–20. 18, 19

[DOOMS ET AL.05] G. Dooms, Y. Deville, and P. Dupont, Workshop on Constraint Based Methods for Bioinformatics CP2005, 2005. 29

[EPPSTEIN 94] D. Eppstein, *Finding the k Shortest Paths*, Tech. report, University of California, Irvine, 1994. 5

[FELL & WAGNER 00] D. A. Fell and A. Wagner, *The small world of metabolism*, Nature America Inc. Metabolic Engineering **18** (2000), 1121–1122. 13, 15

[FORST & SCHULTEN 01] C.V. Forst and K. Schulten, *Phylogenetic Analysis of Metabolic Pathways*, J. Mol. Biol. **52** (2001), 471–489. 13, 15

[FÖRSTER ET AL.03] J. Förster, I. Famili, P. Fu, B.Ø. Palsson, and J. Nielsen, *Genome-Scale Reconstruction of the Saccharomyces cerevisiae Metabolic Network*, Genome Research **13** (2003), 244–253. 24

[GOESMANN ET AL.02] A. Goesmann, M. Haubrock, F. Meyer, J. Kalinowski, and R. Giegerich, *PathFinder: reconstruction and dynamic visualization of metabolic pathways*, Bioinformatics **18** (2002), 124–129. 14, 15

[HOU ET AL.04] B.K. Hou, L.B.M. Ellis, and L.P. Wackett, *Encoding microbial metabolic logic: predicting biodegradation*, J. Ind. Microbiol. Biotechnol. **31** (2004), 261–272. 14, 16, 62

[JEONG ET AL.00] H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai, and A.-L. Barabási, *The large-scale organization of metabolic networks*, Nature **407** (2000), 651–654. 13, 16

[KANEHISA ET AL.02] M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya, *The KEGG databases at GenomeNet*, Nucleic Acids Research **30** (2002), 42–46. 12

[KARP 01] P.D. Karp, *Pathway Databases: A Case Study in Computational Symbolic Theories*, Science **293** (2001), 2040–2044. 12

[KARP ET AL.05] P.D. Karp, C.A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahrén, S. Tsoka, N. Darzentas, V. Kunin, and N. López-Bigas, *Expansion of the BioCyc collection of pathway/genome databases to 160 genomes*, Nucleic Acids Research **33** (2005), 6083–6089. 12

[KÜFFNER ET AL.00] R. Küffner, R. Zimmer, and T. Lengauer, *Pathway analysis in metabolic databases via differential metabolic display*, Bioinformatics **16** (2000), 825–836. 13, 15

[LEMER ET AL.04a] C. Lemer, H. Anerhour, J.M. Maniraja, O. Sand, J. Richelle, and S. Wodak, *The aMAZE database goes public*, ECCB (2004). 21, 23

[LEMER ET AL.04b] C. Lemer, E. Antezana, F. Couche, S. De Keyzer, F. Fays, O. Hubaut, J. Richelle, and S. Wodak, *The Snow system, a tool for the representation and analysis of networks*, ECCB (2004). 21

[MCSHAN ET AL.03] D.C. McShan, S. Rao, and I. Shah, *PathMiner: predicting metabolic pathways by heuristic search*, Bioinformatics **19** (2003), 1692–1698. 14, 16, 19, 62

[NIKITIN ET AL.03] A. Nikitin, S. Egorov, Daraselia N., and I. Mazo, *Pathway studio - the analysis and navigation of molecular networks*, Bioinformatics **19** (2003), 1–3. 18, 19

[PALEY & KARP 06] S.M. Paley and P.D. Karp, *The Pathway Tools cellular overview diagram and Omics Viewer*, Nucleic Acids Research **34** (2006), 3771–3778. 18

[QUACKENBUSH 03] J. Quackenbush, *Microarrays - Guilt by Association*, Science **302** (2003), 240–241. 9

[RAHMAN ET AL.04] S.A. Rahman, P. Advani, R. Schunk, R. Schrader, and D. Schomburg, *Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC)*, Bioinformatics (2004). 14, 15, 16, 20

[RAVASZ ET AL.02] E. Ravasz, A.L. Somera, D.A. Mongru, and Z.N. Oltvai, A.-L. Barabási, *Hierarchical Organization of Modularity in Metabolic Networks*, Science **297** (2002), 1551–1555. 17

[SHANNON ET AL.03] 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** (2003), 2498–2504. 24

[SIBSON 73] R. Sibson, *SLINK: AN optimally efficient algorithm for the single-linkage cluster method*, Computer Journal **16** (1973), 30–34. 31

[SIRAVA ET AL.02] M. Sirava, T. Schaefer, M. Eiglsperger, M. Kaufmann, O. Kohlbacher, E. Bornberg-Bauer, and H.P. Lenhof, *BioMiner - modeling, analyzing, and visualizing biochemical pathways and networks*, Bioinformatics **18** (2002), S219–S230. 14, 15, 19

[SPELLMAN ET AL.98] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, O.P. Brown, D. Botstein, and B. Futcher, *Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization*, Molecular Biology of the Cell **9** (1998), 3273–3297. 30

[THIMM ET AL.04] O. Thimm, O. Blaesing, Y. Gibon, A. Nagel, S. Meyer, P. Krueger, J. Selbig, L.A. Mueller, S.Y. Rhee, and M. Stitt, *MAPMAN: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes*, The Plant Journal **37** (2004), 914–939. 18, 19

[VAN HELDEN ET AL.00] J. van Helden, A. Naim, R. Mancuso, M. Eldridge, L. Wernisch, D. Gilbert, and S. Wodak, *Representing and analysing molecular and cellular function in the computer*, Biol Chem **381** (2000), 921–35. 16, 21

[VAN HELDEN ET AL.01] J. van Helden, D. Gilbert, L. Wernisch, M. Schroeder, and S. Wodak, *Application of Regulatory Sequence Analysis and Metabolic Network Analysis to the Interpretation of Gene Expression Data*, Lecture Notes in Computer Science **2066** (2001), 147–165. 14, 15, 29

[VAN HELDEN ET AL.02] J. van Helden, L. Wernisch, D. Gilbert, and S. Wodak, *Graph-based analysis of metabolic networks*, Ernst Schering Research Foundation Workshop., vol. 38, Springer-Verlag, 2002, pp. 245–274. 16, 26, 29

[WITTIG & DE BEUCKELAER 01] U. Wittig and A. de Beuckelaer, *Analysis and comparison of metabolic pathway databases*, Briefings In Bioinformatics **2** (2001), 126–142. 12