Université Libre de Bruxelles
Faculté des Sciences
Département de Biologie Moléculaire
Service de Bioinformatique des Génomes et des Reséaux

Development, assessment and application of bioinformatics tools for the extraction of pathways from metabolic networks

Karoline Faust

Thèse présentée en vue de l'obtention du grade de Docteur en Sciences

Jury
Président: Prof. Etienne Pays
Vice-président: Prof. Oberdan Leo
Secrétaire: Prof. Anna Maria Marini
Directeur de thèse: Prof. Jacques van Helden
Rapporteur: Prof. Bruno André
Rapporteur: Prof. Tom Lenaerts
Expert extérieur: Prof. Pierre Dupont
Expert extérieur: Prof. Claudine Médigue

Février 2010

Acknowledgement

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 always managed to find time for explanations, discussions and corrections despite of his overfull timetable. Discussions with him, on whatever topic, were always stimulating and always gave new insights. I also appreciate his way of guiding me through the PhD. He left me the independence I need, but at the same time made sure that my research did not go astray.

A big thanks also to Didier Croes, who worked before me on metabolic pathway prediction and on whose work this PhD is based. Although he had left the lab when I started, he took the time to meet me and gave me many explanations and ideas.

I am indebted to all members of the aMAZE team for explaining me their work, answering my questions and giving me access to their code. In particular, I would like to thank Fabian Couche and Frédéric Fays, who introduced me to Eclipse and Java. Christian Lemer gave me several lessons on good programming practice and got me started with ANT. Olivier Hubaut made the useful suggestion of representing seed node groups by pseudo nodes.

I would like to thank all the current and former BiGre members for their kindness. I was warmly welcomed and enjoyed a lot the lab's social activities, such as the Christmas parties and the girls' evenings with Morgane, Gipsi and Alejandra. In particular I would like to thank Raphaël, whom I bothered regularly with computer problems and whose ongoing efforts simply keep the computers of the lab running. A big thanks also to Morgane, the good fairy of the lab. She helped me an uncounted number of times with various issues. Sylvain was a great colleague whose cheerful personality and reliability made it a pleasure to work with him. Thanks to Gipsi for numerous scientific discussions and relaxing chats and to Alejandra for introducing me to Mexican drinking and eating habits. It was great to embark on the adventure of parenthood together with so many colleagues in the same year, among them Jean-Valery, Rekin's, Sylvain, Matthieu and Gipsi. My thanks also goes to Myriam, Marc, Olivier, Ariane and Fernanda for their support and help with different problems and questions. A special thanks to Raul, with whom I had many interesting discussions on various topics.

I am grateful to Bruno André, the president of my PhD commission, and the other commission members, Marcelline Kaufmann and Gianluca Bontempi, who gave useful advice, support and suggestions throughout my thesis. Thanks as well to Patrice Godard, who gave me access to his microarray data.

Pierre Dupont patiently explained me the kWalks algorithm and made numerous contributions to this work. I am grateful for his ideas, his exceptional communication skills and his friendliness.

I would like to acknowledge Pierre Schaus, Jean-Noël Monette and their supervisor Yves Deville, who helped me during the initial stages of my PhD. Thanks as well to Grégoire Dooms, who introduced me to (CP)Graph.

I also wish to thank the members of my jury and especially Claudine Médigue, who accepted to take the trouble to come from Paris to Brussels.

A big thanks to Didier Gonze and my family for their unfailing support and love. Thanks also to Didier's family and Didier for several babysitting sessions, without which I could not have finished this thesis. Last but not least I would like to thank Sophie for her smiles, for her sweetness, for her presence in my life.

Abstract

Genes can be associated in numerous ways, e.g. by co-expression in micro-arrays, co-regulation in operons and regulons or co-localization on the genome. Association of genes often indicates that they contribute to a common biological function, such as a pathway. The aim of this thesis is to predict metabolic pathways from associated enzyme-coding genes. The prediction approach developed in this work consists of two steps: First, the reactions are obtained that are carried out by the enzymes coded by the genes. Second, the gaps between these seed reactions are filled with intermediate compounds and reactions. In order to select these intermediates, metabolic data is needed. This work made use of metabolic data collected from the two major metabolic databases, KEGG and MetaCyc. The metabolic data is represented as a network (or graph) consisting of reaction nodes and compound nodes. Intermediate compounds and reactions are then predicted by connecting the seed reactions obtained from the query genes in this metabolic network using a graph algorithm.

In large metabolic networks, there are numerous ways to connect the seed reactions. The main problem of the graph-based prediction approach is to differentiate biochemically valid connections from others. Metabolic networks contain hub compounds, which are involved in a large number of reactions, such as ATP, NADPH, H_2O or CO_2. When a graph algorithm traverses the metabolic network via these hub compounds, the resulting metabolic pathway is often biochemically invalid.

In the first step of the thesis, an already existing approach to predict pathways from two seeds was improved. In the previous approach, the metabolic network was weighted to penalize hub compounds and an extensive evaluation was performed, which showed that the weighted network yielded higher prediction accuracies than either a raw or filtered network (where hub compounds are removed). In the improved approach, hub compounds are avoided using reaction-specific side/main compound annotations from KEGG RPAIR. As an evaluation showed, this approach in combination with weights increases prediction accuracy with respect to the weighted, filtered and raw network.

In the second step of the thesis, path finding between two seeds was extended to pathway prediction given multiple seeds. Several multiple-seed pathay prediction approaches were evaluated, namely three Steiner tree solving heuristics and a random-walk based algorithm called kWalks. The evaluation showed that a combination of kWalks with a Steiner tree heuristic applied to a weighted graph yielded the highest prediction accuracy.

Finally, the best perfoming algorithm was applied to a microarray data set, which measured gene expression in *S. cerevisiae* cells growing on 21 different compounds as sole nitrogen source. For 20 nitrogen sources, gene groups were obtained that were significantly over-expressed or suppressed with respect to urea as reference nitrogen source. For each of these 40 gene groups, a metabolic pathway was predicted that represents the part of metabolism up- or down-regulated in the presence of the investigated nitrogen source.

The graph-based prediction of pathways is not restricted to metabolic networks. It may be applied to any biological network and to any data set yielding groups of associated genes, enzymes or compounds. Thus, multiple-end pathway prediction can serve to interpret various high-throughput data sets.

Abbreviations

AL Average path length

ADP Adenosine Diphosphate

ATP Adenosine Triphosphate

EC Enzyme Commission

E. coli Escherichia coli

EM Elementary Mode

FN False Negative

FP False Positive

GABA Gamma-aminobutyric acid

GML Graph Modelling Language

HQL Hibernate Query Language

IQR Interquartile Range

KEGG Kyoto Encyclopedia of Genes and Genomes

NAD Nicotinamide Adenine Dinucleotide

NADP Nicotinamide Adenine Dinucleotide Phosphate

NCR Nitrogen Catabolite Repression

NeAT Network Analysis Tools

ORF Open Reading Frame

OWL Web Ontology Language

PPV Positive Predictive Value

REA Recursive Enumeration Algorithm

S. cerevisiae Saccharomyces cerevisiae

SGD Saccharomyces Genome Database

SMILES Simplified Molecular Input Line Entry System

TCA cycle Tricarboxylic Acid cycle, also known as Krebs or citric acid cycle

TP True Positive

Publication list

K. Faust and J. van Helden
Predicting metabolic pathways by subnetwork extraction
Methods in Molecular Biology - Bacterial Molecular Networks, Submitted.

K. Faust, P. Dupont, J. Callut and J. van Helden
Pathway discovery in metabolic networks by subgraph extraction
Submitted.

K. Faust, D. Croes and J. van Helden
In response to "Can sugars be produced from fatty acids? A test case for pathway analysis tools"
Bioinformatics, vol. 25, pp. 3202-3205, 2009.

K. Faust, D. Croes and J. van Helden
Metabolic Pathfinding Using RPAIR Annotation
Journal of Molecular Biology, vol. 388, pp. 390-414, 2009.

K. Faust, J. Callut, P. Dupont and J. van Helden
Inference of pathways from metabolic networks by subgraph extraction
Proceedings of the second International Workshop on Machine Learning in Systems Biology 2008.

S. Brohée, K. Faust, G. Lima-Mendez, O. Sand, R. Janky, G. Vanderstocken, Y. Deville and J. van Helden
NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways
Nucleic Acids Research, vol. 36, pp. W444-W451, 2008.

K. Faust[1], S. Brohee, Gipsi Lima-Mendez, G. Vanderstocken and J. van Helden
Network Analysis Tools: from biological networks to clusters and pathways
Nature Protocols, vol. 3, pp. 1616-1629, 2008.

[1] K. Faust and S. Brohée contributed equally to this publication.

Contents

1 Introduction

1.1 Motivation

High-throughput experiments such as microarrays allow to measure gene expression in different conditions at a genomic scale. Interpretation of these data is however a challenging task. One approach commonly applied to the interpretation of microarray data has been termed "Guilt by association" [137]. 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.

Genes may not only be "guilty of association" by co-expression, but also by co-regulation in operons and regulons, co-occurrence in phylogeny, co-localization in the genome or in other ways. In all these cases, it is of interest to identify the biological module or pathway in which the "guilty" genes are involved.

In order to identify pathways from associated genes, many software tools simply map the genes on a set of pre-defined pathways [34, 159, 123, 84, 64, 1]. An example for pathway mapping is given in Figure 1.1.

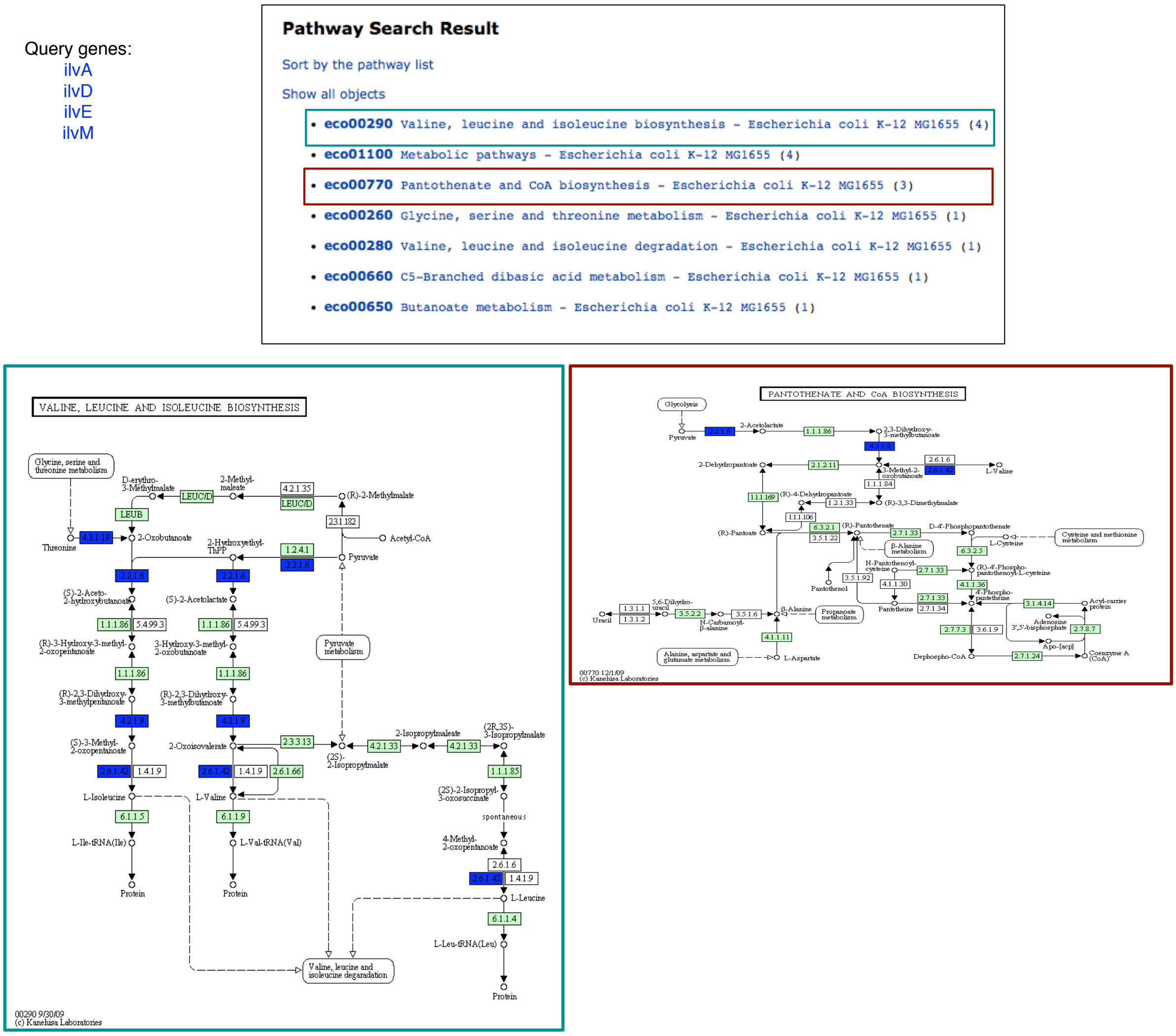
This mapping approach has several shortcomings:

- It is only applicable to organisms with correspondences to known pathways.
- It does not deal well with genes mapping to several pathways.
- It fails to find variants of pre-defined pathways.
- It is unable to uncover novel pathways from known components (e.g. known reactions, compounds or protein interactions).

A major reason for these drawbacks is the inability of the mapping approaches to take the interconnection of biological pathways into account. The accumulation of more and more biological data triggered recently a shift in data representation from modules and pathways towards whole networks. Networks have the advantage to account for the interconnection of pathways and to enable a series of interesting analyses. However, predicting relevant biological pathways from networks instead of pre-defined pathways poses a challenge.

1.2 Goal of the thesis

The goal of this thesis is to predict biochemically relevant metabolic pathways from metabolic networks and groups of associated enzyme-coding genes. The metabolic network is assembled



Query genes:
ilvA
ilvD
ilvE
ilvM

Pathway Search Result

Sort by the pathway list

Show all objects

- **eco00290** Valine, leucine and isoleucine biosynthesis - Escherichia coli K-12 MG1655 (4)
- **eco01100** Metabolic pathways - Escherichia coli K-12 MG1655 (4)
- **eco00770** Pantothenate and CoA biosynthesis - Escherichia coli K-12 MG1655 (3)
- **eco00260** Glycine, serine and threonine metabolism - Escherichia coli K-12 MG1655 (1)
- **eco00280** Valine, leucine and isoleucine degradation - Escherichia coli K-12 MG1655 (1)
- **eco00660** C5-Branched dibasic acid metabolism - Escherichia coli K-12 MG1655 (1)
- **eco00650** Butanoate metabolism - Escherichia coli K-12 MG1655 (1)

Figure 1.1: Illustration of pathway mapping. The query genes ilvD, ilvE, ilvM and ilvA (all members of the same operon in *E. coli*) were mapped to KEGG pathway maps [86], using the KEGG tool: "Color Objects in KEGG Pathways". The genes occur in six pathway maps (not counting the overview map). A selection of two maps is shown, each of which contains more than one query gene. Reactions associated to query genes are colored in blue, *E. coli* specific reactions are colored in green.

from a metabolic database, whereas the genes can come from a variety of data sources (co-expression, co-regulation, co-occurrence in phylogenetic profiles etc.). From the enzyme-coding genes, reactions are obtained, which serve as *seeds* for pathway prediction. Figure 1.2 depicts a flow chart of metabolic pathway prediction.

The first step to reach this goal consisted in the development of a new algorithm and the adaptation of existing algorithms to the prediction of pathways from metabolic networks. In a second step, the prediction accuracy of these algorithms was evaluated on a large set of annotated metabolic pathways. During this evaluation, the impact of various parameters (such as network properties or the integration of main/side compound annotations from the RPAIR database [96]) on the accuracy was assessed. Finally, the best-performing algorithm was applied to extract pathways from co-expressed gene groups obtained from a microarray data set.

1.3 Biological background: Metabolism

The classical biochemistry textbook [13] defines metabolism as follows: "Metabolism is essentially a linked series of chemical reactions that begins with a particular molecule and converts it into some other molecule or molecules in a carefully defined fashion."

Chemical reactions involved in metabolism will from now on be termed *metabolic reactions* or *reactions*. They act on molecules, which will be termed *compounds* throughout this work.

Metabolic reactions can be subdivided into two basic categories: *Anabolic reactions* synthesize molecules from smaller building blocks whilst consuming energy, whereas *catabolic reactions* break down large molecules to generate energy and building blocks required by anabolic reactions.

Another classification of metabolism is based on the importance of compounds for the survival of an organism. *Primary compounds*, such as glucose 6-phosphate or ATP, are involved in the maintenance of the basic functions of life (growth, development, reproduction). *Secondary compounds* are not required for these basic tasks, but are needed in specific conditions, such as defense against parasites (e.g. antibiotics produced by fungi).

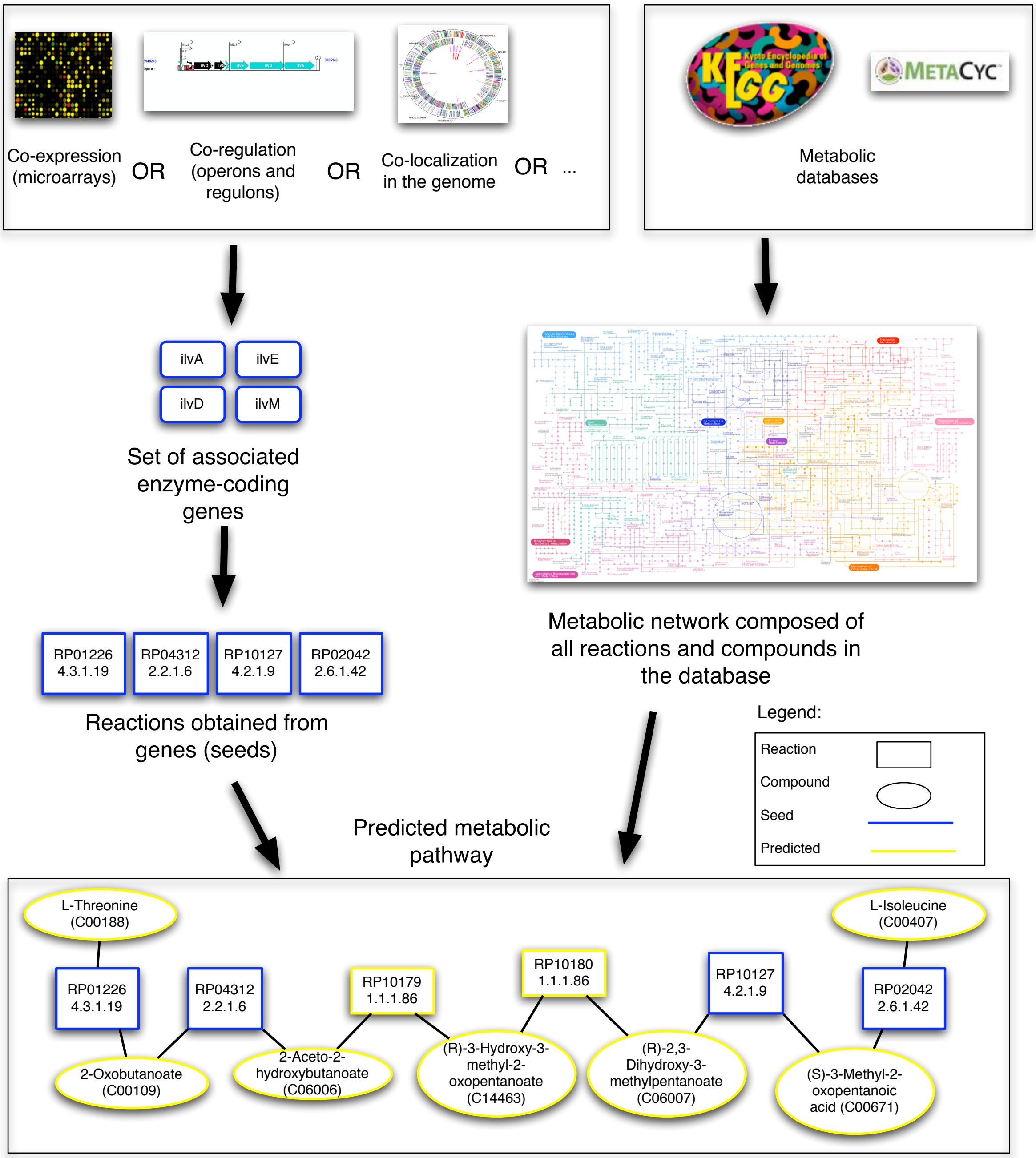
As will be discussed in detail in section 1.5, metabolism can be represented as a network. This representation allows to define core and peripheral metabolism. This classification loosely corresponds to the traditional classification into primary and secondary compounds and will be presented in section 1.6.4.

A typical cell is composed of thousands of compounds and reactions. For instance EcoCyc (version 13.1) [90] lists 1,415 enzymes, 1,784 reactions and 1,753 compounds for *Escherichia coli*.

In the following, the major concepts of metabolism will be discussed in more detail.

1.3.1 Compounds

The chemical structure of a compound can be represented in a variety of ways. Most important are the *sum formula*, which lists the numbers of different atoms contained in the compound and the *structural formula*, which shows how the atoms are arranged. For bioinformatics, the



Co-expression (microarrays)
OR
Co-regulation (operons and regulons)
OR
Co-localization in the genome
OR
...
Metabolic databases
ilvA
ilvE
ilvD
ilvM
Set of associated enzyme-coding genes
RP01226 4.3.1.19
RP04312 2.2.1.6
RP10127 4.2.1.9
RP02042 2.6.1.42
Reactions obtained from genes (seeds)
Metabolic network composed of all reactions and compounds in the database
Legend:
Reaction
Compound
Seed
Predicted
Predicted metabolic pathway
L-Threonine (C00188)
L-Isoleucine (C00407)
RP01226 4.3.1.19
RP04312 2.2.1.6
RP10179 1.1.1.86
RP10180 1.1.1.86
RP10127 4.2.1.9
RP02042 2.6.1.42
2-Oxobutanoate (C00109)
2-Aceto-2-hydroxybutanoate (C06006)
(R)-3-Hydroxy-3-methyl-2-oxopentanoate (C14463)
(R)-2,3-Dihydroxy-3-methylpentanoate (C06007)
(S)-3-Methyl-2-oxopentanoic acid (C00671)

Figure 1.2: Flow chart of metabolic pathway prediction. Genes can be associated in several ways, e.g. by co-expression, co-regulation or co-localization. The first step is to obtain a set of reactions from the associated, enzyme-coding genes. These reactions are then submitted to the metabolic pathway prediction tool, which takes two inputs: 1) the set of *seed* reactions or compounds and 2) a metabolic network constructed from a publicly available metabolic database, such as KEGG [86]. A metabolic pathway is predicted by connecting the seeds in the metabolic network. Image sources: global KEGG network and KEGG symbol were obtained from the KEGG database homepage [86], the MetaCyc symbol from the MetaCyc home page [22]. The operon image was obtained from RegulonDB [61] and the genome image from the Comprehensive Microbial Resources [36]. The microarray image was taken from the Liverpool Microarray facility homepage (with kind permission from the University of Liverpool).

representation of a compound by a string of characters is also relevant. One such string representation is the *SMILES* notation [168]. Figure 1.3 illustrates these different representations on the example of D-glucose.

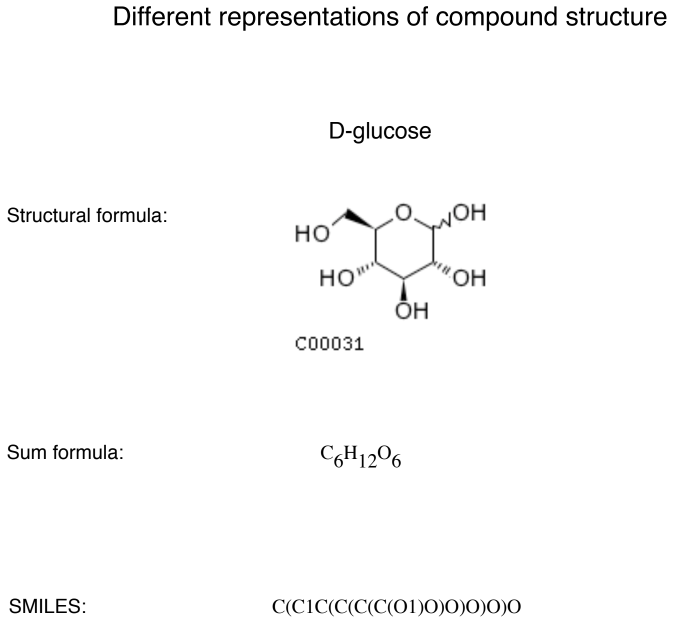


Figure 1.3: The chemical structure, the sum formula and the SMILES representation of the compound D-glucose are shown. The structure image was taken from KEGG [86].

Compounds take on different roles in metabolism. For instance, many authors distinguish between main and side compounds of a pathway. *Main compounds* "carry" a major part of the carbon atoms through the reactions of the pathway. They form the "backbone" of pathways, as Karp put it [88] and are also called intermediate compounds. Typical *side compounds* act as donors/acceptors of energy or electrons, such as ATP/ADP and NAD(P)$^+$/NAD(P)H or as donors/acceptors of functional groups such as tetrahydrofolic acid. The distinction between main and side compounds poses problems discussed in section 1.7.2. In addition, other atom types than carbon may be relevant in a pathway, e.g. the sulfur incorporation pathway contains intermediates such as sulfide that do not contain carbon.

Side compounds such as ATP/ADP or NAD(P)$^+$/NAD(P)H and small inorganic compounds such as H_2O or CO_2 are involved in many reactions and are therefore called *ubiquitous* or *hub* compounds. As will be discussed in section 1.10.3, appropriate treatment of hub compounds is necessary to predict metabolic pathways accurately.

In this thesis, only small molecules and their reactions are taken into account. Polymers such as DNA, proteins or RNA and their reactions are neglected.

1.3.2 Reactions

Reactions convert a set of input compounds called *substrates* into a set of *products*. Substrates and products together are also named *reactants*. Figure 1.4 depicts an example reaction with two substrates and two products.

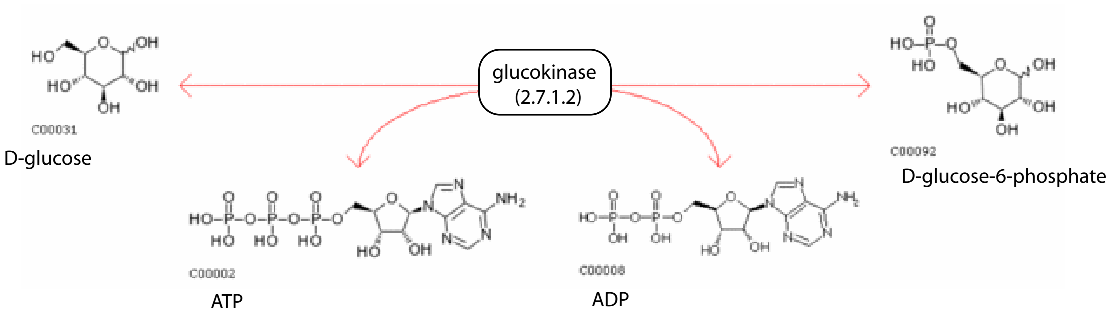


Figure 1.4: Example of a metabolic reaction (KEGG identifier R00299). The substrates D-glucose and ATP are converted into the products D-glucose-6-phosphate and ADP. The reaction is catalyzed by the enzyme glucokinase, which has the EC number 2.7.1.2 assigned to it. The direction of this reaction is not specified by this illustration. However, in physiological conditions, this reaction is irreversible, since it consumes energy by hydrolyzing ATP. The compound structure images were taken from KEGG [86].

A reaction can proceed in two directions. Consider for example reaction `A + B ↔ C`. It can proceed from A and B towards C (*forward direction*) or from C towards A and B (*reverse direction*). In this thesis, the following notation is adobted: If the arrow in a reaction equation points from left to right, compounds on the left-hand side are considered as substrates and compounds on the right-hand side as products. If it points from right to left, right-hand side compounds are substrates and left-hand side compounds are products.

The direction of a reaction depends on its change in Gibbs free energy ΔG. If $\Delta G = 0$, the reaction is at equilibrium and neither the forward nor the reverse direction is preferred. If $\Delta G > 0$, the forward direction is unfavorable and the reaction cannot occur spontaneously (endergonic reactions). If $\Delta G < 0$, the forward direction is preferred and the reaction may occur spontaneously (exergonic reactions).

ΔG is a function of the concentrations of the reactants, the temperature and the standard free energy change ΔG_0 of the reaction: $\Delta G = \Delta G_0 + RT lnQ$, with the reaction quotient $Q = \frac{[product_1]^c ... [product_n^d]}{[substrate_1^a] ... [substrate_m^b]}$, where n is the number of products and m the number of substrates of the reaction and a,b, c and d are stoichiometric coefficients of the substrates and products. The standard free energy change is defined as $\Delta G_0 = -RT lnK_{eq}$, with the equilibrium constant $K_{eq} = Q_{equilibrium}$. $Q_{equilibrium}$ refers to the reactant concentrations in equilibrium. The standard conditions are defined as follows: one molar concentration of reactants, one atmosphere pressure and for metabolic reactions pH 7.0.

Reactions that are far from equilibrium ($\Delta G \neq 0$) are also termed irreversible reactions, because the preference for one reaction direction is so great that the other one can be neglected. Since ΔG depends on reactant concentrations and the temperature, a reaction that is physiological irreversible in one organism may be reversible in another one. For instance, the temperature in thermophilic organisms is different from the temperature in mesophilic organisms, thus their ΔG values also differ, even if reactant concentrations are the same.

These considerations affect the representation of reactions in metabolic networks comprising several organisms.

1.3.3 Enzymes

Enzymes are biomolecules that catalyze reactions. Most enzymes are proteins, but RNA may also act as an enzyme (ribozyme). Importantly, an enzyme does not change the equilibrium constant of a reaction; it only accelerates its rate. Not all reactions are catalyzed by enzymes: Some occur *spontaneously*, e.g. the conversion from L-glutamate gamma-semialdehyde into water and (S)-1-pyrroline-5-carboxylate (MetaCyc identifier: SPONTPRO-RXN).

Enzymes are hierarchically classified based on the reactions they catalyze [121]. The classification scheme consists of four levels, each represented by a separate digit and ordered from most generic to most specific. The first level comprises the following 6 categories:

1. *Oxidoreductases* catalyze oxidation/reduction reactions.

2. *Transferases* catalyze the transfer of functional groups.

3. *Hydrolases* catalyze the hydrolyzation of a substrate into two products.

4. *Lyases* catalyze the non-hydrolytic addition or removal of atom groups from substrates.

5. *Isomerases* catalyze isomeric changes within a single compound.

6. *Ligases* join together two molecules under ATP consumption.

Figures 1.5 and 1.6 give an example for each category of the first level.

For instance, an enzyme with the EC number 5.3.1.9 belongs to the isomerases (category 5). More specifically, it is an intramolecular oxidoreductase (5.3), which interconverts aldoses and ketoses (5.3.1). The fourth digit is an index to distinguish enzymes acting on different reactants (glucose-6-phosphate and fructose-6-phosphate in this case).

Relating enzymes to reactions

A many-to-many relationship exists between enzymes and reactions, meaning that one enzyme may catalyze several reactions and that one reaction may be catalyzed by several enzymes.

Multifunctional enzymes such as Aro1p in *Saccharomyces cerevisiae* may carry out several subsequent reactions (5 in the case of Aro1p). A multifunctional enzyme possesses multiple functional sites, which enable compound *channeling*.

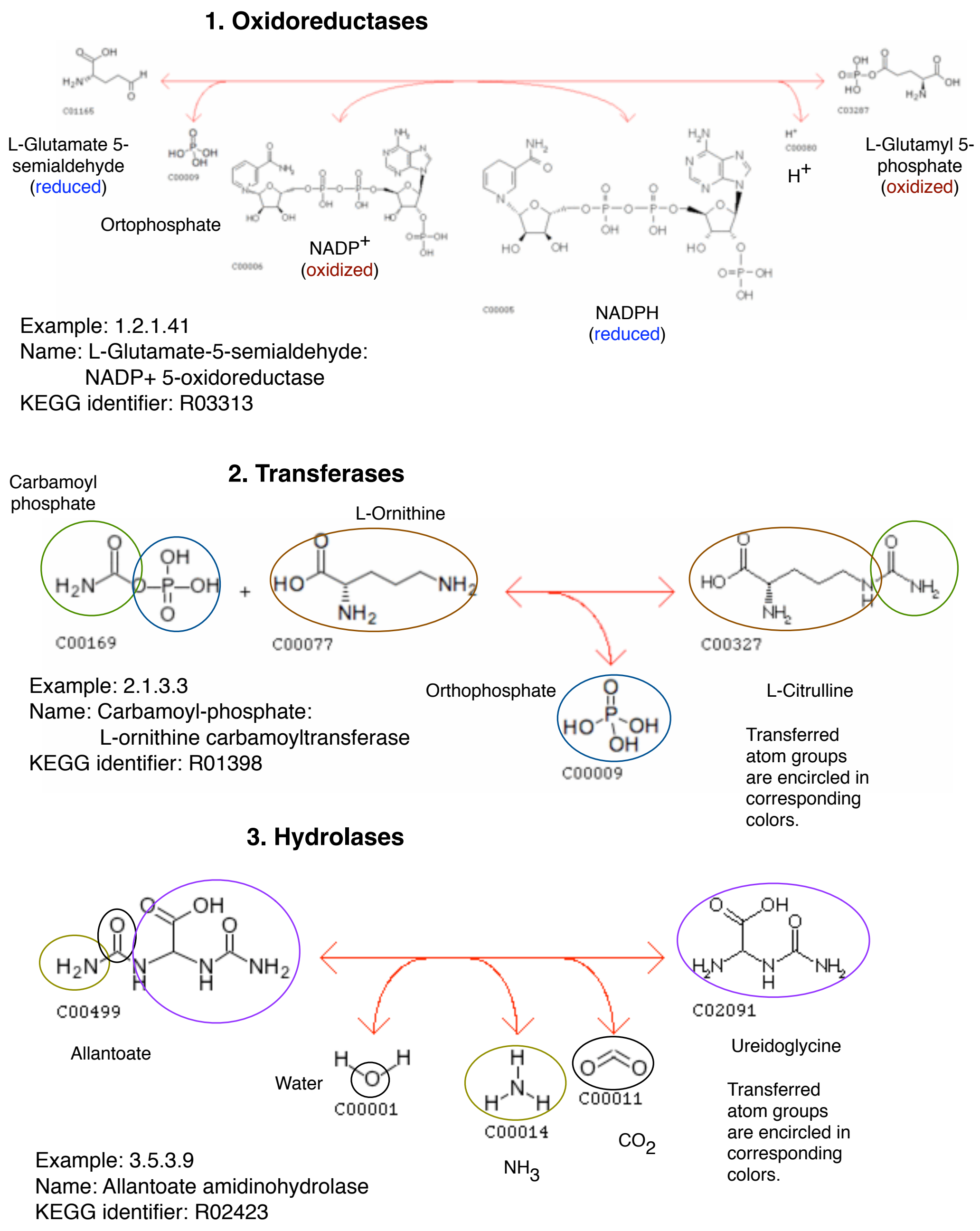
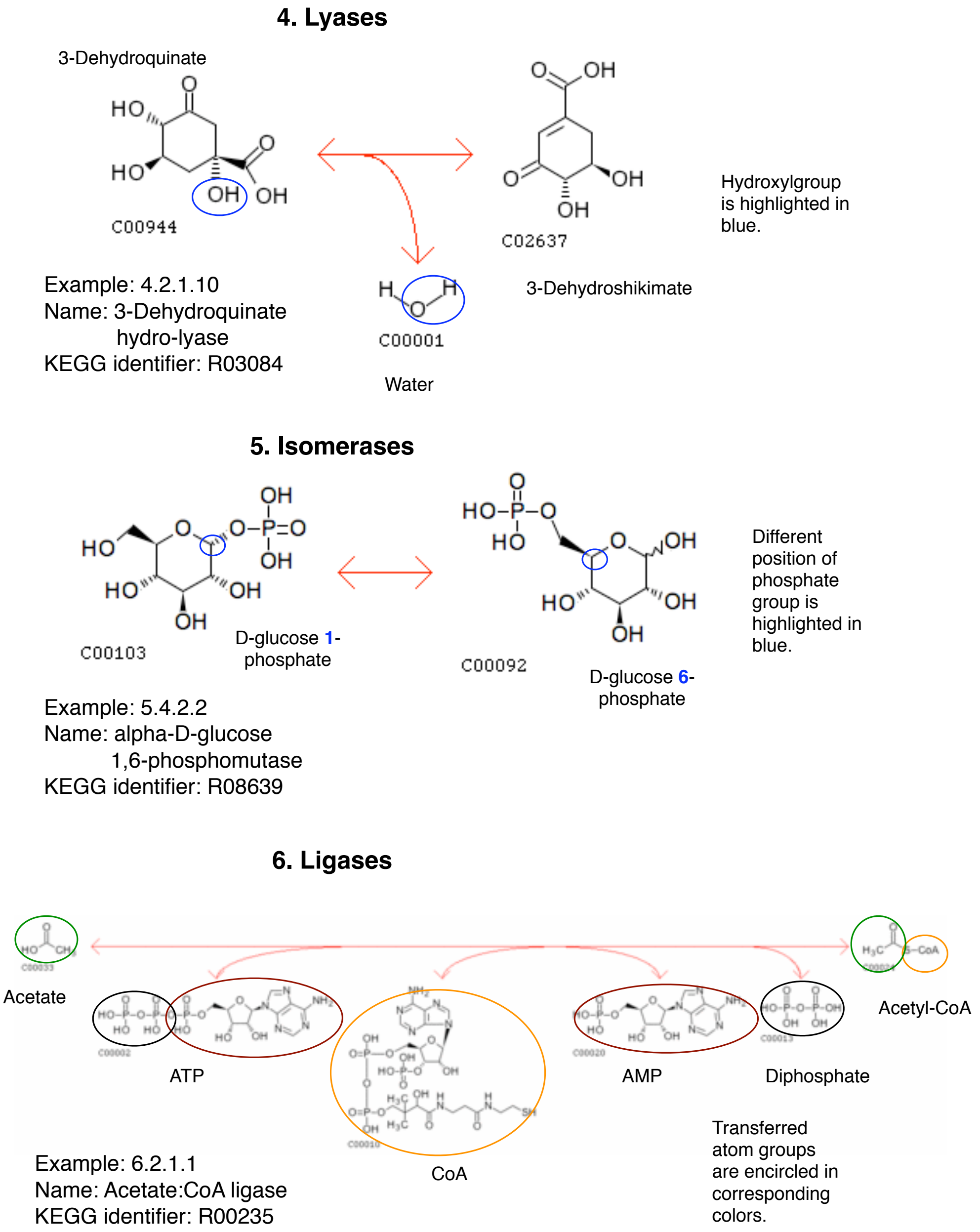


Figure 1.5: Examples for the first, second and third category of the first level of the enzyme classification scheme. The compound structure images were taken from KEGG [86].



Example: 4.2.1.10
Name: 3-Dehydroquinate hydro-lyase
KEGG identifier: R03084

Hydroxylgroup is highlighted in blue.

Example: 5.4.2.2
Name: alpha-D-glucose 1,6-phosphomutase
KEGG identifier: R08639

Different position of phosphate group is highlighted in blue.

Example: 6.2.1.1
Name: Acetate:CoA ligase
KEGG identifier: R00235

Transferred atom groups are encircled in corresponding colors.

Figure 1.6: Examples for the fourth, fifth and sixth category of the first level of the enzyme classification scheme. The compound structure images were taken from KEGG [86].

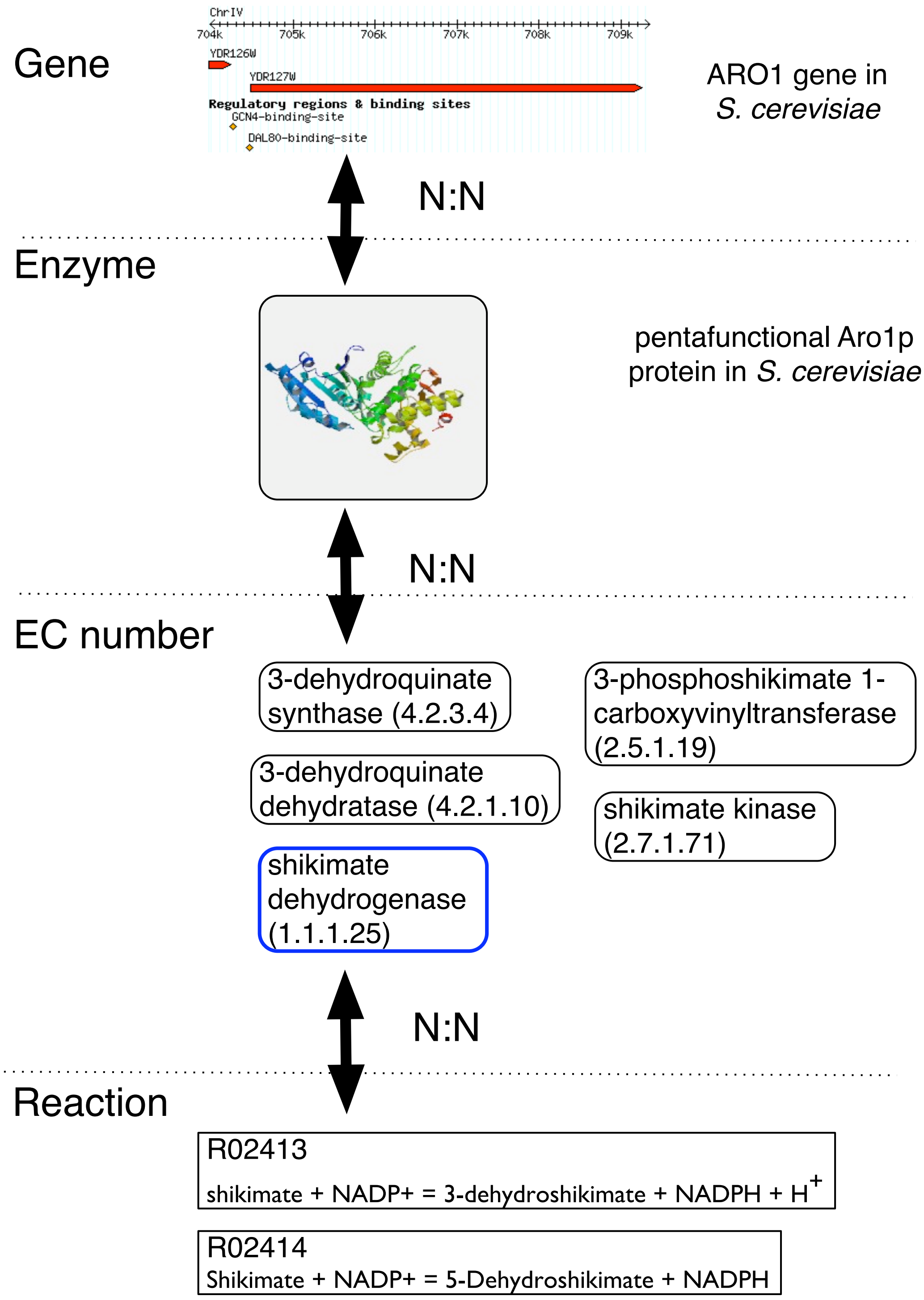

Gene
ChrIV
704k
705k
706k
707k
708k
709k
YDR126W
YDR127W
Regulatory regions & binding sites
GCN4-binding-site
DAL80-binding-site
ARO1 gene in
S. cerevisiae
N:N
Enzyme
pentafunctional Aro1p
protein in S. cerevisiae
N:N
EC number
3-dehydroquinate synthase (4.2.3.4)
3-dehydroquinate dehydratase (4.2.1.10)
shikimate dehydrogenase (1.1.1.25)
3-phosphoshikimate 1-carboxyvinyltransferase (2.5.1.19)
shikimate kinase (2.7.1.71)
N:N
Reaction
R02413
shikimate + NADP+ = 3-dehydroshikimate + NADPH + H+
R02414
Shikimate + NADP+ = 5-Dehydroshikimate + NADPH

Figure 1.7: This Figure illustrates the many-to-many relationships between genes and reactions with the pentafunctional ARO1 gene from *S. cerevisiae* as an example. Altogether, this gene is associated to six reactions. Only the reactions associated to shikimate dehydrogenase are shown. Image sources: The image of the ORF is taken from SGD [74], the image of the protein structure from PDB [14].

When compounds are channeled, they are not released from the enzymes but passed from one enzyme to another one or, as in the case of Aro1p, from one catalytic site to another one, thus increasing the efficiency of metabolism.

Isoenzymes catalyze the same reaction, e.g. in *Escherichia coli* three different aspartate kinases catalyze the conversion from L-aspartate to L-aspartate-4-phosphate. Each is regulated in another way, thus isoenzymes allow the cell to fine-tune metabolic reactions.

Furthermore, a many-to-many relationship exists between EC numbers and reactions, because reactions with the same EC number may differ by their substrates and one reaction may be catalyzed by various catalytic mechanisms (corresponding to different EC numbers). For instance, homoserine dehydrogenase with EC number 1.1.1.3 converts L-homoserine into L-aspartate 4-semialdehyde. There are two reactions associated to this EC number, having either NAD^+ or $NADP^+$ as a co-substrate. Another example is EC number 1.1.1.23, which is associated to two reactions: The first converts histidinol to histidinal and the second histidinal to histidine.

Finally, a many-to-many relationship exists between genes and enzymes. Several genes may code for different sub-units of one enzyme and several enzymes may be synthesized from one gene via alternative splicing. Figure 1.7 illustrates the many-to-many relationships between an enzyme-coding gene and its associated reactions.

Regulation of enzyme-coding genes

The activity of enzymes may be regulated on several levels, including the transcriptional level (regulation of the enzyme-coding gene's expression) and the post-transcriptional level (inhibitors and activators binding to the enzyme). As this thesis deals with the interpretation of a set of associated genes, the focus here is on the transcriptional level. Enzyme-coding genes involved in a common pathway are frequently grouped in *operons* and *regulons*. An *operon* is a part of the genome that contains a set of genes that are controlled by common regulatory elements, and that are transcribed together from a common *promoter* (the binding site of the RNA polymerase). Figure 1.8 shows an example of an operon. A *regulon* is defined as "a set of genes subject to regulation of one and only one regulator." (RegulonDB, [61]). In contrast to an operon, the genes of a regulon are not necessarily under the control of a single promoter. A classical example of a regulon is the arginine repressor of *E. coli* [108].

Operons and regulons allow the cell to switch on or off an entire pathway in response to environmental signals.

1.4 Metabolic databases

Metabolic information is available in the classical metabolic textbooks, in the biochemical literature and more recently in metabolic databases.

The two most important generic metabolic databases are KEGG [86] and BioCyc [22]. Other metabolic databases, such as Reactome [165] (human metabolism) and UM-BBD [50] (microbial degradation pathways) are also relevant, but more specialized.

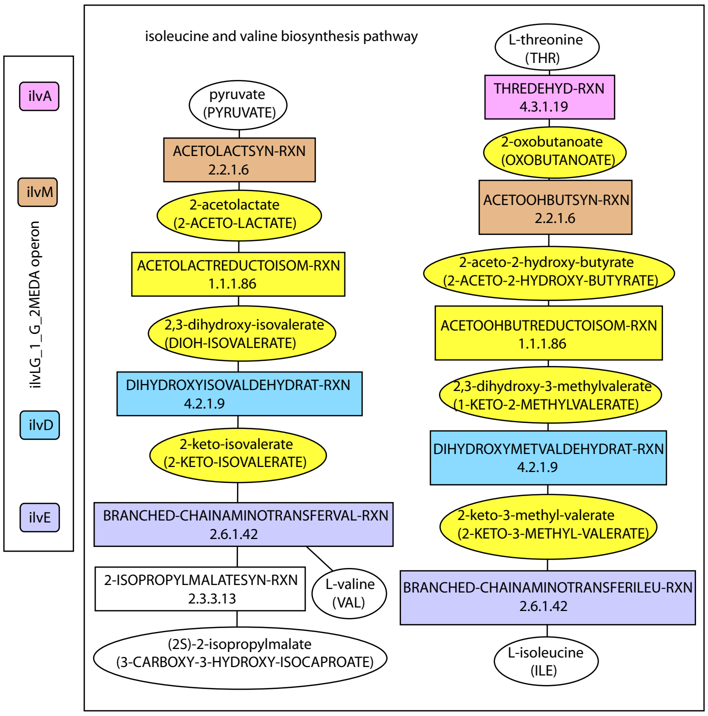

ilvA
ilvM
ilvLG_1_G_2MEDA operon
ilvD
ilvE
isoleucine and valine biosynthesis pathway
pyruvate (PYRUVATE)
ACETOLACTSYN-RXN 2.2.1.6
2-acetolactate (2-ACETO-LACTATE)
ACETOLACTREDUCTOISOM-RXN 1.1.1.86
2,3-dihydroxy-isovalerate (DIOH-ISOVALERATE)
DIHYDROXYISOVALDEHYDRAT-RXN 4.2.1.9
2-keto-isovalerate (2-KETO-ISOVALERATE)
BRANCHED-CHAINAMINOTRANSFERVAL-RXN 2.6.1.42
2-ISOPROPYLMALATESYN-RXN 2.3.3.13
L-valine (VAL)
(2S)-2-isopropylmalate (3-CARBOXY-3-HYDROXY-ISOCAPROATE)
L-threonine (THR)
THREDEHYD-RXN 4.3.1.19
2-oxobutanoate (OXOBUTANOATE)
ACETOOHBUTSYN-RXN 2.2.1.6
2-aceto-2-hydroxy-butyrate (2-ACETO-2-HYDROXY-BUTYRATE)
ACETOOHBUTREDUCTOISOM-RXN 1.1.1.86
2,3-dihydroxy-3-methylvalerate (1-KETO-2-METHYLVALERATE)
DIHYDROXYMETVALDEHYDRAT-RXN 4.2.1.9
2-keto-3-methyl-valerate (2-KETO-3-METHYL-VALERATE)
BRANCHED-CHAINAMINOTRANSFERILEU-RXN 2.6.1.42
L-isoleucine (ILE)

Figure 1.8: The ilvLG_1G_2MEDA operon of *E. coli* contains eight genes, four of which are involved in the synthesis of valine and isoleucine. Each of the enzyme-coding genes is associated to metabolic reactions (the color code allows to match each gene to its respective reactions). Compounds and reactions not associated to the ilvLG_1G_2MEDA operon are colored in yellow.

KEGG and BioCyc cover many organisms and store pathways that are either collected from the literature or predicted with automated or semi-automated metabolic reconstruction procedures [87, 114]. Both consist of a collection of databases (KEGG: KEGG LIGAND, KEGG RPAIRS, KEGG GLYCAN ...), (BioCyc: MetaCyc, EcoCyc, HumanCyc, ...).

KEGG and BioCyc differ in the way they organize the metabolic data. In BioCyc, organism-specific metabolic pathways are dynamically drawn, whereas KEGG shows static maps that unite all known reactions involved in a common "theme" as defined by the KEGG team, such as purine metabolism. Organism-specific reactions in these maps can be highlighted upon mouse-click.

KEGG and BioCyc have different strengths and weaknesses [169, 103]. The advantage of KEGG for metabolic pathway prediction is the manual annotation of the reactant pairs of a reaction and their roles, which are provided by the RPAIR database [96]. Its disadvantage is the absence of organism-specific pathways. Because a KEGG map is not conceived as a pathway, but rather as a union of all known reactions belonging to a common "metabolic theme", it cannot serve as reference for the evaluation of pathway prediction. BioCyc documents better than KEGG its sources (literature, experimental evidence) and separates clearer between predicted and annotated pathways. Instead of "a union of reactions" it displays organism-specific pathways at different levels of detail, which in contrast to KEGG include the side compounds. The relationship between EC numbers and reactions is also less ambiguous than in KEGG. However, reactant pair annotation as in KEGG RPAIR is absent from BioCyc. Table 1.1 summarizes the differences between KEGG and BioCyc that are relevant for pathway prediction, whereas Figure 1.9 shows an example that illustrates the different pathway concept of KEGG and BioCyc.

The differences between KEGG maps and BioCyc pathways illustrate that it is not always clear where to draw the border between metabolic pathways. This touches upon the problem of metabolic pathway definition, which will be discussed in detail in section 1.7.

1.5 Mapping of metabolism onto a network

In order to predict metabolic pathways, metabolic data has to be represented in a structured fashion that makes the application of prediction algorithms possible.

Metabolism has been represented using stoichiometric matrices, graphs, rule sets, first-order logic and other formalisms (see section 1.10.3). Since the pathway prediction techniques presented in this work are based on the extraction of relevant parts from graphs, the representation of metabolism as a graph will be discussed in more detail.

A *graph* is a mathematical abstraction of connected objects and consists of nodes (also called vertices) and edges, which connect the nodes (See appendix A for a brief introduction to graph theory). In this thesis, the terms "network" and "graph" are more or less used as synonyms. *Network* refers to a set of interconnected biological objects (e.g. compounds and reactions in a metabolic network), whereas *graph* refers to the formal representation of a network.

It is not trivial to map metabolism onto a graph in a meaningful way. Some network representations suffer from important drawbacks, and are therefore less suited for the prediction of

Table 1.1: Differences between KEGG and BioCyc that are relevant for metabolic pathway prediction

Property	KEGG	BioCyc
Pathway display	Clickable, static maps	Dynamically drawn, clickable pathways
Pathway variants	Maps merge all known variants of a pathway	Each variant is a separate pathway
Organism specificity	Organism-specific parts of maps can be highlighted	Pathway variants are organism-specific
Side compounds	Maps do not include side compounds	Pathways include side compounds, which are marked as such
EC number-reaction-mapping	EC number reaction relationships are sometimes ambiguous	EC number reaction relationships are rarely ambiguous
Direct gene-reaction-relationships	Stored in KGML files, but not accessible via web interface or API	Stored in biopax files and accessible via web interface
Reactant pair annotation	KEGG RPAIR	Absent
Predicted versus annotated pathways	Status not clearly indicated	Predicted pathways well separated from annotated pathways (Tier 1, 2 and 3)
Documentation of data sources	Links to literature are sparse	Literature sources well referenced

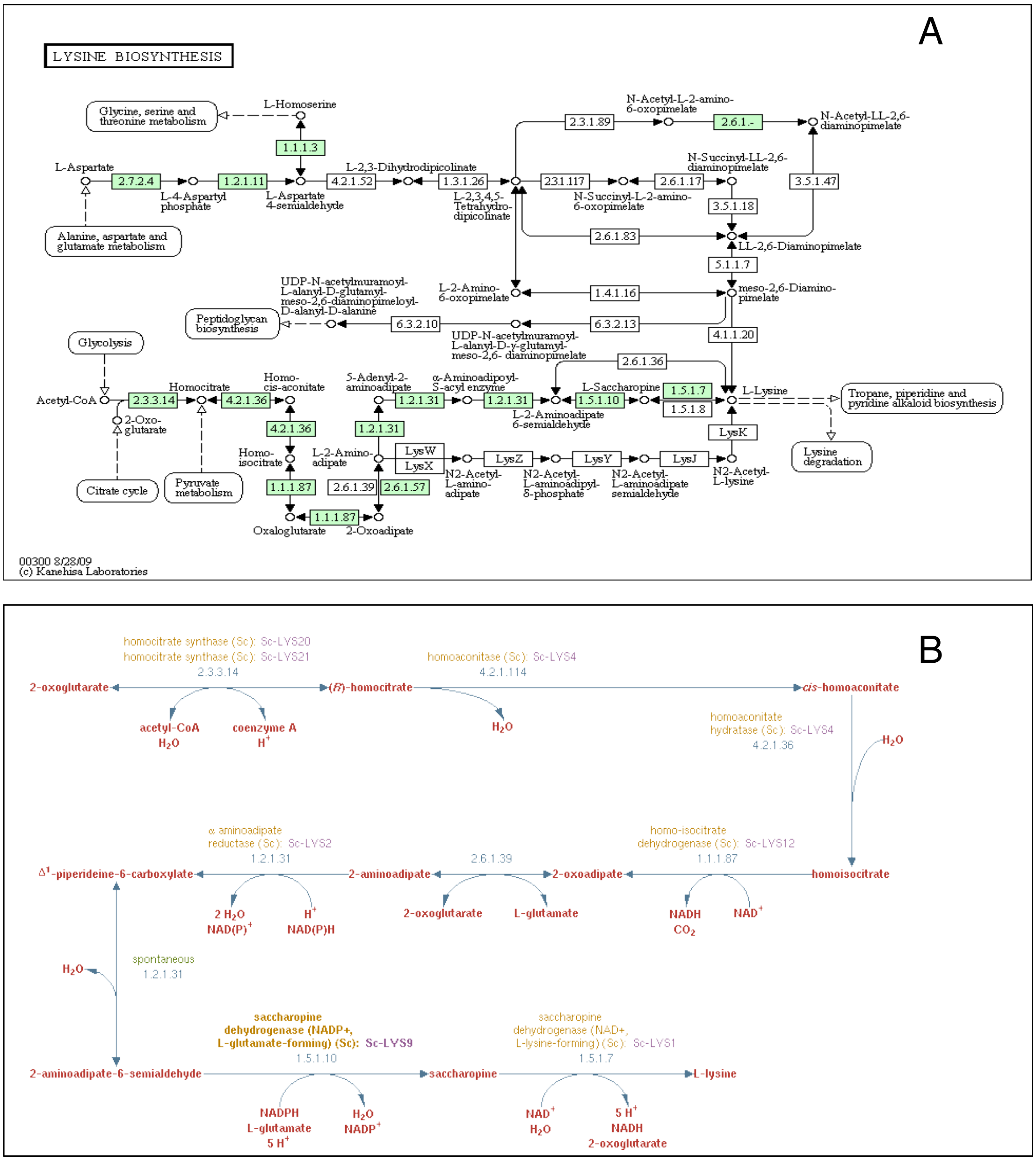

A
LYSINE BIOSYNTHESIS
Glycine, serine and threonine metabolism
L-Homoserine
1.1.1.3
L-Aspartate
2.7.2.4
1.2.1.11
L-4-Aspartyl phosphate
L-Aspartate 4-semialdehyde
4.2.1.52
L-2,3-Dihydrodipicolinate
1.3.1.26
L-2,3,4,5-Tetrahydro-dipicolinate
Alanine, aspartate and glutamate metabolism
2.3.1.89
N-Acetyl-L-2-amino-6-oxopimelate
2.6.1.-
N-Acetyl-LL-2,6-diaminopimelate
2.3.1.117
2.6.1.17
N-Succinyl-LL-2,6-diaminopimelate
3.5.1.47
N-Succinyl-L-2-amino-6-oxopimelate
3.5.1.18
2.6.1.83
LL-2,6-Diaminopimelate
5.1.1.7
UDP-N-acetylmuramoyl-L-alanyl-D-glutamyl-meso-2,6-diaminopimeloyl-D-alanyl-D-alanine
L-2-Amino-6-oxopimelate
1.4.1.16
meso-2,6-Diamino-pimelate
Peptidoglycan biosynthesis
6.3.2.10
6.3.2.13
4.1.1.20
UDP-N-acetylmuramoyl-L-alanyl-D-γ-glutamyl-meso-2,6-diaminopimelate
2.6.1.36
Glycolysis
Acetyl-CoA
2.3.3.14
Homocitrate
4.2.1.36
Homo-cis-aconitate
5-Adenyl-2-aminoadipate
α-Aminoadipoyl-S-acyl enzyme
1.2.1.31
1.2.1.31
1.5.1.10
L-Saccharopine
1.5.1.7
1.5.1.8
L-Lysine
Tropane, piperidine and pyridine alkaloid biosynthesis
L-2-Aminoadipate 6-semialdehyde
2-Oxo-glutarate
4.2.1.36
1.2.1.31
LysK
Homo-isocitrate
L-2-Amino-adipate
LysW
LysX
LysZ
LysY
LysJ
Lysine degradation
Citrate cycle
Pyruvate metabolism
1.1.1.87
2.6.1.39
2.6.1.57
N2-Acetyl-L-amino-adipate
N2-Acetyl-L-aminoadipyl-δ-phosphate
N2-Acetyl-L-aminoadipate semialdehyde
N2-Acetyl-L-lysine
1.1.1.87
Oxaloglutarate
2-Oxoadipate
00300 8/28/09
(c) Kanehisa Laboratories
B
homocitrate synthase (Sc): Sc-LYS20
homocitrate synthase (Sc): Sc-LYS21
2.3.3.14
homoaconitase (Sc): Sc-LYS4
4.2.1.114
2-oxoglutarate
(R)-homocitrate
cis-homoaconitate
acetyl-CoA
H2O
coenzyme A
H+
H2O
homoaconitate hydratase (Sc): Sc-LYS4
4.2.1.36
H2O
α aminoadipate reductase (Sc): Sc-LYS2
1.2.1.31
2.6.1.39
homo-isocitrate dehydrogenase (Sc): Sc-LYS12
1.1.1.87
Δ1-piperideine-6-carboxylate
2-aminoadipate
2-oxoadipate
homoisocitrate
2 H2O
NAD(P)+
H+
NAD(P)H
2-oxoglutarate
L-glutamate
NADH
CO2
NAD+
H2O
spontaneous
1.2.1.31
saccharopine dehydrogenase (NADP+, L-glutamate-forming) (Sc): Sc-LYS9
1.5.1.10
saccharopine dehydrogenase (NAD+, L-lysine-forming) (Sc): Sc-LYS1
1.5.1.7
2-aminoadipate-6-semialdehyde
saccharopine
L-lysine
NADPH
L-glutamate
5 H+
H2O
NADP+
NAD+
H2O
5 H+
NADH
2-oxoglutarate

Figure 1.9: Representation of lysine biosynthesis in *S. cerevisiae* in KEGG (A) and MetaCyc (B). In KEGG, the lysine biosynthesis pathway map contains all reactions known to be involved in lysine biosynthesis. Thus, not a species-specific pathway is displayed, but species-specific parts of the map are highlighted (in green). In MetaCyc, six variants of lysine biosynthesis are stored as separate pathways, each of which is specific to a set of organisms. The pathway shown in B is the lysine biosynthesis pathway IV (MetaCyc identifier LYSINE-AMINOAD-PWY).

metabolic pathways than others [164].

- *Compound-centered networks.* A node represents a compound and an edge represents a reaction connecting two compounds. Since many reactions involve more than one substrate and/or product, several edges represent the same reaction. A graph traversal algorithm may thus cross the same reaction several times, in the worst case connecting one of its substrates with another one or one product with another one.

- *Reaction-centered networks.* A node represents a reaction and an edge represents a compound that is the product of one reaction and the substrate of another one. This representation faces the same problem as the compound-centered network: a compound involved in several reactions is represented by several edges, thus a graph traversal algorithm may cross the same compound several times.

- *Bipartite networks.* Bipartite networks consist of two node sets: one represents compounds and the other reactions. A special case of bipartite networks are *Petri nets*, where the two nodes sets are named *places* and *transitions*, respectively. Places can be marked by *tokens* and *firing rules* can be defined on the transitions that describe the consumption of tokens from input places and the production of tokens in output places. When applied to metabolism, compounds are treated as places and reactions as transitions [98]. Another special case of this network type is the *and-or graph* employed in [131], where reactions are represented by *and-nodes* and compounds by *or-nodes*. Bipartite networks avoid the problems of compound-centered and reaction-centered networks and in addition allow to search paths between compounds and/or reactions [164] and have therefore been selected for this thesis.

- *Hypergraphs.* In hypergraphs, an edge may connect more than two nodes. In principle, compound-centered and reaction-centered hypergraphs could be used, but so far, only the compound-centered hypergraph has been mentioned in the metabolic pathway prediction literature [113], with compounds as nodes and reactions as *hyperarcs* (directed *hyperedges*). The stoichiometric matrix employed in flux balance analysis is mathematically equivalent to a (compound-centered) hypergraph ([93]). Despite of their recent recommendation in [93], hypergraphs have a disadvantage for pathway prediction: It is not as easy as in bipartite graphs to predict pathways for an input combining compounds and reactions.

Figure 1.10 summarizes the different network representations.

Another issue is the representation of reaction directionality in a network. As discussed in section 1.3.2, all reactions are reversible in principle. However, organism-specific metabolic networks should account for physiologically irreversible reactions. A precondition for the representation of irreversible reactions is a directed network. Directed metabolic networks also avoid another pitfall: Their directedness prevents a graph traversal algorithm to go from one substrate to another substrate or from one product to another product of the same reaction. For instance, consider the example reaction in Figure 1.4. If the algorithm could go from one substrate to the second, the resulting pathway would suggest the synthesis of ATP from D-glucose within one step, which is biochemically impossible. In a directed bipartite metabolic network, reversible reactions can be represented by including two nodes; one for each reaction direction (see Figure 1.10 F). Irreversible reactions can then be represented by including only one direction node. To prevent a graph traversal algorithm to go twice through the same reaction, forward and reverse direction node have to exclude each other mutually, i.e. they cannot both appear in the same path [31, 32]. Thus, a XOR (exclusive OR) relationship exists between the forward and reverse direction of a reaction. If the direction nodes would not be mutually exclusive, the following pathway could be predicted from the reaction shown in Figure 1.4: `D-glucose` → `2.7.1.2_forward` → `D-glucose 6-phosphate` → `2.7.1.2_reverse` → `ATP`. This pathway falsely suggests that ATP can be synthesized from D-glucose within two reaction steps.

In this thesis, metabolic networks were constructed from all reactions and compounds present in a metabolic database and are thus not organism-specific. In these generic networks, all irreversible reactions were represented as reversible for two reasons: (1) to avoid conflicting reaction directions in case a reaction proceeds in one direction in one organism and in the other direction in another organism and (2) to take into account the fact that all reactions are potentially reversible (see section 1.3.2). However, the tools developed during this thesis can as well deal with networks containing irreversible reactions.

To summarize: Metabolic networks in this thesis are, if not stated otherwise, directed bipartite networks in which each reaction is represented by two mutually exclusive nodes: one for its forward and one for its reverse direction.

1.6 Properties of metabolic networks

The representation of metabolism as a network allows to quantify a number of topological properties.

1.6.1 Power law and small world property

In [82] topological properties of metabolic networks from 43 different species have been measured. First, the authors plotted the distribution of *compound node degree* frequencies on a logarithmic scale, where the degree of a compound is the number of reactions it is involved in. They found that this distribution follows a *power-law* $P(k) \approx k^{-\gamma}$, where k is the compound node degree and $P(k)$ is the probability of degree k in the network. Figure 1.11 shows a compound node degree distribution for KEGG data. The power-law is indicative of a *scale-free* property of the network, i.e. "any part of the scale-free network is stochastically similar to the whole network, and parameters are assumed to be independent of the system size" [91]. This particular topology is not displayed by random networks.

Second, Jeong et al. measured the *network diameter*, which they define as "the shortest biochemical pathway averaged over all pairs of substrates" [82] and which is around three for the investigated metabolic networks. From the small network diameter, Jeong and coworkers deduce a *small world* property of the metabolic network, which states that each node can be reached from each other node within a few steps.

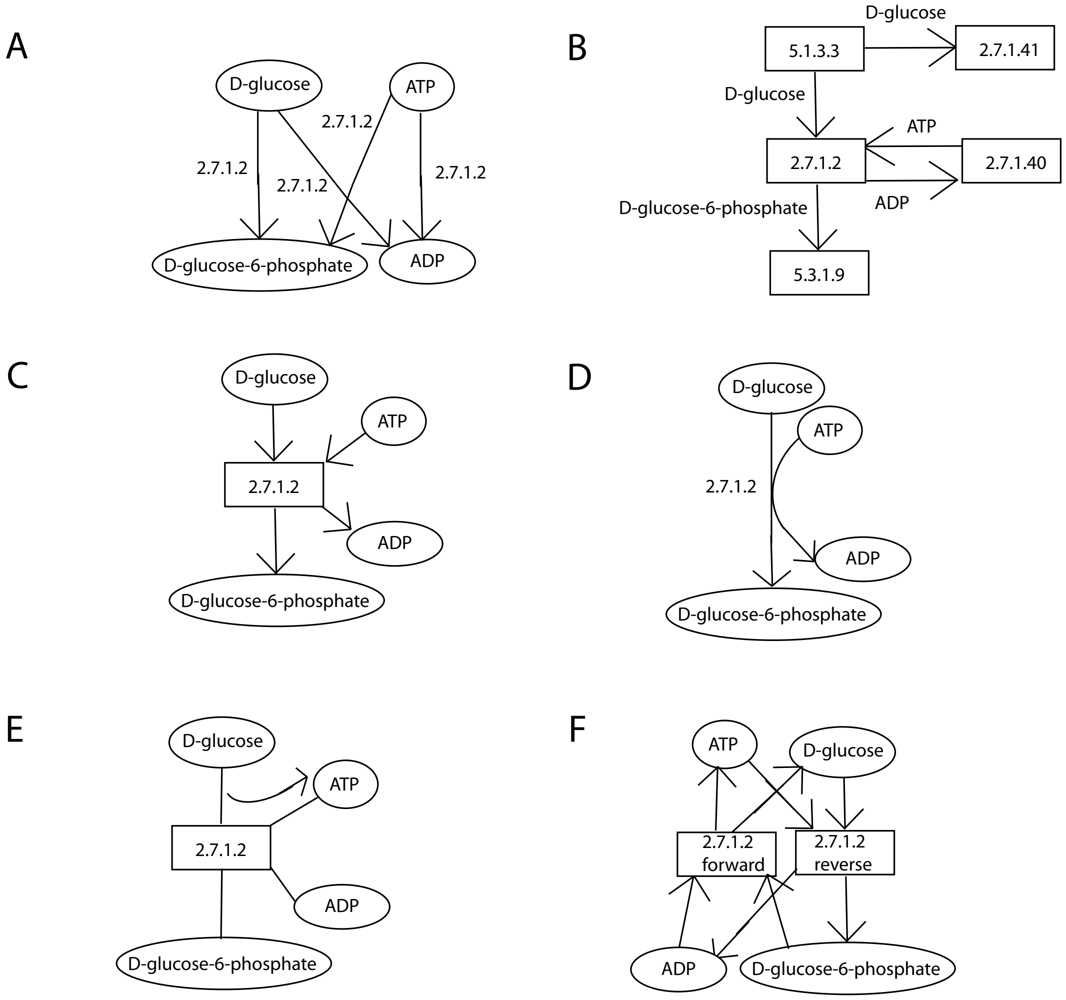


Figure 1.10: Figure A gives an example for the compound-centered network, Figure B for the reaction-centered network, Figure C for a bipartite network and Figure D for a hypergraph. Note that all networks are directed. Figure E illustrates the problem of undirected networks: a graph traversal algorithm can easily go from one substrate to another one (in this case from D-glucose to ATP) or from one product to another one. Figure F illustrates how each reaction direction is represented by its own node in the directed bipartite network. The two nodes 2.7.1.2_forward and 2.7.1.2_reverse are mutually exclusive, i.e. they cannot both appear in the same path.

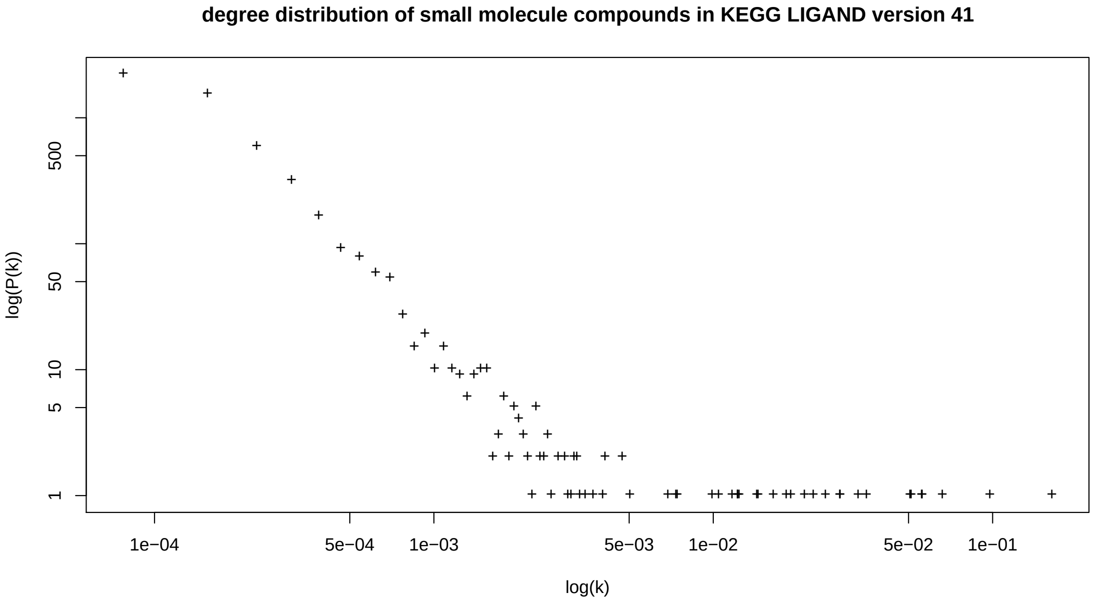


Figure 1.11: An example of the kind of plot introduced by [82]. The distribution of compound node degree probabilities $P(k)$ in a small molecule network constructed from KEGG LIGAND version 41.0 is plotted on a logarithmic scale. The probability of a degree is estimated by its frequency, i.e. the ratio between the number of nodes having this degree and the total number of nodes in the network. This plot differs from the one shown in [82] inasmuch as the degree is not separated into in-degree (number of incoming arcs) and out-degree (number of outgoing arcs), the node degrees are not binned and the metabolic data comes from another source. It can be seen that a linear function (that is a power law in logarithmic scale) does not describe well the two tails of this distribution.

It should be noted that some authors employ a different definition of network diameter, which is: "... the path length of the longest pathway among all the shortest pathways [107]". The network diameter definition of Jeong and coworkers corresponds rather to the *average path length* (abbreviated AL, also called characteristic path length), which is defined by Watts and Strogatz as the "number of edges in the shortest path between two nodes, averaged over all pairs of nodes" [41].

Jeong et al. postulated that the small world property of metabolic networks is due to the presence of hub compounds, i.e. compounds involved in a large number of reactions. To give an example of hub compounds, Table 1.2 lists the top ten hub compounds for the small molecule metabolic networks used in this thesis, namely the KEGG LIGAND network, KEGG RPAIR network (both version 41.0) and MetaCyc network (version 11). It is of note that the KEGG LIGAND and MetaCyc lists are similar, but differ from the KEGG RPAIR list. The reason for this difference will be explained in section 1.10.3.

Table 1.2: Top ten hub compounds of the three networks used in this thesis.

KEGG LIGAND version 41.0	KEGG RPAIR version 41.0	MetaCyc version 11.0
H_2O	H_2O	H_2O
H^+	ATP	O_2
O_2	NH_3	NADPH
NADP	S-Adenosylmethionine	NADP
NADPH	CO_2	NAD
NAD	pyruvate	NADH
NADH	glutamate	ATP
ATP	acetyl-CoA	CO_2
CO_2	O_2	phosphate
phosphate	oxoglutarate	diphosphate

The two properties claimed by Jeong and coworkers (power-law degree distribution and small world) have been questioned by a number of authors. The small world property will be criticized in section 1.10.3. Khanin and Wit computed the degree distributions of several biological networks and measured the goodness of fit of different functions to those distributions. Their results demonstrated that these biological networks do not follow a power law distribution [91]. In a recent review, the power law and the small world property were declared to be "myths of network biology" [104], because they are not statistically valid (power law) or even due to the computation of shortest paths that are biochemically invalid (small world, see section 1.10.3).

1.6.2 Modularity and hierarchical organization

Several authors point out the modularity of metabolic networks [140, 128] or describe algorithms to partition a metabolic network into smaller units [144, 68].

In [140], the modularity of the *Escherichia coli* metabolic network is measured with the *cluster coefficient*. The node-specific cluster coefficient C_i is a function of the fraction of realized connections among all possible connections between a node and its group of neighbors. It is defined as

$$C_i = \frac{2n}{k_i(k_i - 1)} \tag{1.1}$$

where i is a node index, n is the number of direct links between the k nearest neighbours of node i and N is the number of nodes in the network. The cluster coefficient C is then obtained as the average over all node-specific cluster coefficients. The cluster coefficient indicates that the metabolic network of *E. coli* is highly modular.

Ravasz et al. [140] pointed out a contradiction between the high modularity of metabolic networks on the one hand and the presence of hub compounds connecting all nodes on the other hand. To resolve this contradiction, they suggest that metabolic networks are hierarchically structured. In a hierarchically structured network, several small modules form larger modules, thus creating a hierarchy of nested modules. The modules have many intra-module connections, but only few inter-module connections, which explains how a modular network can give rise to a power law distribution of node degrees. To check this hypothesis, Ravasz and colleagues defined a topological overlap matrix, where a value of 1 indicates that two compounds are connected to the same neighbors and a value of 0 indicates that two compounds do not share neighbors. Then, they applied a hierarchical clustering algorithm to this topological overlap matrix and identified modules at various levels of distance. In many cases, these modules corresponded well to biochemical units (e.g. purine metabolism).

1.6.3 Bow-tie shape

Several authors have commented on the *bow-tie* shape of metabolism [33, 175]. This special shape arises because hundreds of nutrients (the left fan of the bow-tie) are catabolized into a small set of precursors (the knot of the bow-tie), from which the hundreds of building blocks needed by the cell are synthesized (the right fan of the bow-tie).

It has been stated that this particular structure has evolved to allow rapid adaptation, to tolerate perturbations and to ease regulation [33].

1.6.4 Core and periphery

In section 1.3, the classification of metabolism into core and peripheral has been mentioned. In [2], this classification receives a graph-theoretical basis. Core reactions are defined as connected sets of reactions that are active in all tested environments, whereas peripheral reactions are only activated in specific environments. The effect of different environments on reaction activity is simulated with flux balance analysis, which calculates flux distributions through metabolic networks given an objective function. The core, which corresponds to the "knot" of the bow-tie, was found to be conserved across different organisms, whereas the periphery (the two fans of the bow-tie) alters considerably, reflecting adaptation to different environments [128].

In [126] the idea of a conserved core is extended to find the minimal set of *essential* yeast genes, that is the minimal number of genes required by the cell to survive in a nutrient-rich medium. The authors conclude that the major part of dispensable genes are important in specific environments and only 20% of yeast genes are essential. In a nutrient-rich medium, yeast cells survive with a minimal core metabolic network, but as soon as nutrients are lacking, "peripheral" pathways need to be activated.

1.7 Metabolic pathway definition

A precise definition of metabolic pathways eases their accurate prediction. However, it is not as easy as one might think to precisely define what is meant by "metabolic pathway", a fact that is also underlined by the large number of existing definitions.

In the following, different definitions are listed, approximately sorted from more general to more specific. A definition is more general than another one if it covers more pathways than the other one.

The definitions have been conceived with different questions in mind and it is therefore difficult to find criteria to compare them.

A number of metabolic pathways are described in the literature, which are confirmed by various experiments. Since these pathways represent biochemical knowledge, they should be covered by a good pathway definition. It should also be possible to experimentally validate pathways satisfying the definition in question. Finally, a good definition should prohibit pathways that violate basic principles of biochemistry.

Many of the pathway definitions listed below are tailored to specific pathway validation experiments (discussed in section 1.9) and linked to a particular metabolic pathway prediction approach (see section 1.10.3). Table 1.3 compares different definitions according to these critera.

Table 1.3: Summary of metabolic pathway definitions.

Metabolic pathway definition	Coverage of known metabolic pathways	Consideration of hub compounds	Selected experimental validation techniques	Problems of definition for metabolic pathway prediction
Classical/ topological	Yes, all	No	Atom tracing, in vitro reconstitution, mutant construction, ...	This definition provides no distinction between biochemically irrelevant and relevant pathways.
Atom-flow based	If carbon only is considered, sulfur incorporation and similar pathways are not covered, but definition may be extended to other atom types.	Hub compounds are avoided by atom tracing.	Atom tracing experiments.	Atom mappings are required to apply this definition in practice. They may be obtained computationally or manually.
Feasibility-based	Yes, with appropriate contents of substrate compound set A and auxiliary compound set S.	Yes, with auxiliary compound set.	In vitro reconstitution.	In vivo experimental validation may be difficult, because compounds not specified as substrates or auxiliary compounds should not interfere. Assignment of compounds to sets A and S may require prior

				knowledge of the pathways to be predicted.
Functional	Not all reference pathways are defined on the basis of their regulation.	Yes, implicitly.	Manipulation of activators or inhibitors affecting the expression of genes in the pathway, measurements of gene co-expression in a variety of conditions.	Does not cover all the reference pathways. Requires knowledge of regulation.
Stoichiometric	Some reference pathways contain unbalanced internal compounds, e.g. TCA cycle [132].	Yes, by treating hub compunds as external compounds.	Measurement of fluxes with atom tracing techniques. Measurement of concentrations of selected compounds and of growth rate for wild type versus knock-out mutants.	Does not cover all the reference pathways. External compounds have to be assigned manually. EM analysis assumes additionally that metabolism is in steady state (e.g. compound concentrations remain constant at a relevant time scale)
Chemical organization theory	With appropriate choices for in- and outflows, most reference pathways should be covered.	Yes, hub compounds can be treated as external by adding in- and outflows.	In vitro reconstitution.	External compounds have to be assigned manually. In vivo experimental validation may be difficult.

At this point, the difference between "path" and "pathway" should be clarified. A path is a graph-theoretical concept that refers to a linear sequence of nodes (see Appendix A for details), whereas the term (metabolic) pathway refers to a set of interconnected reactions involved in common biological function. In contrast to paths, pathways may be branched.

1.7.1 Classical definition of metabolic pathways

In the words of the classical biochemistry textbook Nelson and Cox [117], a metabolic pathway is a “sequence of enzyme-catalysed reactions by which a living organism transforms an initial source compound into a final target compound.”

This definition has to be extended to cover branched and cyclic pathways (such as aromatic amino acid biosynthesis and TCA cycle, see Figure 1.12) and to take into account spontaneous reactions. It may be reformulated thus:

“A metabolic pathway is a sequence of enzyme-catalyzed or spontaneous reactions by which a living organism transforms an initial set of source compounds into a final set of target compounds, where source and target compound sets may overlap.”

As discussed in section 1.5, metabolism may be represented as a graph. A graph-theoretical formulation of the classical definition is for example:

“A metabolic network is a directed reaction graph with substrates as vertices and directed, labeled edges denoting reactions between substrates catalyzed by enzymes. A metabolic pathway is a special case of a metabolic network with distinct start and end points, initial and terminal vertices, respectively, and a unique path between them” [59]. Note that this definition implies that metabolism is represented by a directed compound-centered network. It is however easy to adapt to bipartite networks. In addition, this definition needs the same modifications as Nelson and Cox’s definition in order to account for spontaneous reactions and cyclic pathways.

More generally, a graph-theoretical formulation of the classical pathway definition could be:

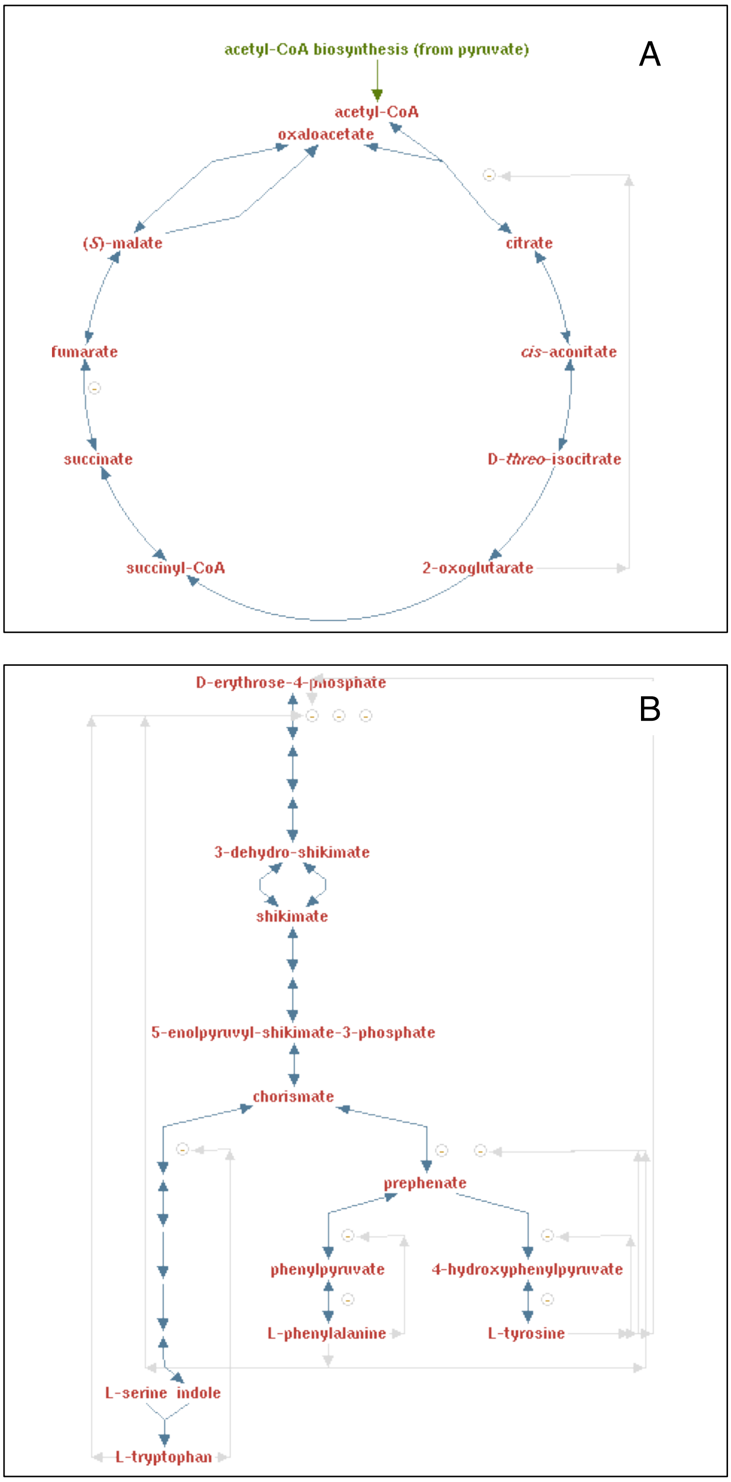


Figure 1.12: Examples for a cyclic pathway (TCA cycle, Figure A) and a branched pathway (aromatic amino acid biosynthesis, Figure B). The pathway images were taken from MetaCyc [22].

"A metabolic pathway is a connected subgraph of the metabolic graph."

Since the classical definition can be expressed entirely in a graph theoretical form, without the need of additional concepts, it could be also called the *topological* definition of metabolic pathways.

Küffner et al. applied the topological definition consequently and enumerated paths between glucose and pyruvate in a network constructed from KEGG, Brenda [26] and ENZYME [9]. They found no less than 500,000 paths [98]! This illustrates well the problem of combinatorial explosion that a pathway prediction algorithm is faced with.

Not all of these paths are relevant biochemically. Consider for example the path D-glucose → 2.7.1.2 → ADP → 2.7.1.40 → pyruvate shown in Figure 1.13. This path suggests that pyruvate can be synthesized from D-glucose in two steps with ADP as an intermediate compound. Such a pathway is biochemically impossible, because the structures of D-glucose and ADP and the structures of ADP and pyruvate are so different that no enzyme can carry out all the required atomic re-arrangements, additions and removals in one reaction.

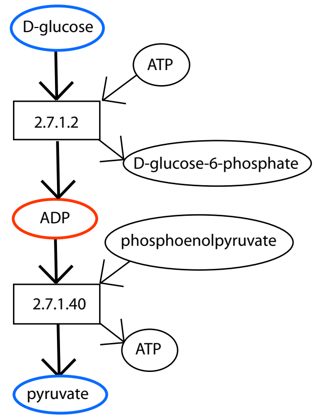


Figure 1.13: Pathway illustrating the problem of the classical metabolic pathway definition, namely that not all sequences of reactions represent biochemically acceptable pathways. In this example pathway, the two reactions are connected via ADP, which is a side compound that does not carry atoms from glucose to pyruvate.

Thus, the classical definition needs refinement to exclude irrelevant pathways.

The rule-based and weighted network pathway prediction approaches (see 1.10.3) make use of the classical definition with some restrictions, thereby manually or automatically excluding irrelevant pathways.

1.7.2 Atom-flow based definitions of metabolic pathways

Atom-flow based pathway definitions take into account that a metabolic pathway transfers atom groups from a source compound to a target compound. Figure 1.14 illustrates the transfer of atom groups in the example reaction shown in Figure 1.4.

Arita is credited with the invention of the atom-flow based pathway definition, which he phrases as follows: "A metabolic pathway (pathway for short) from metabolite X to Y is

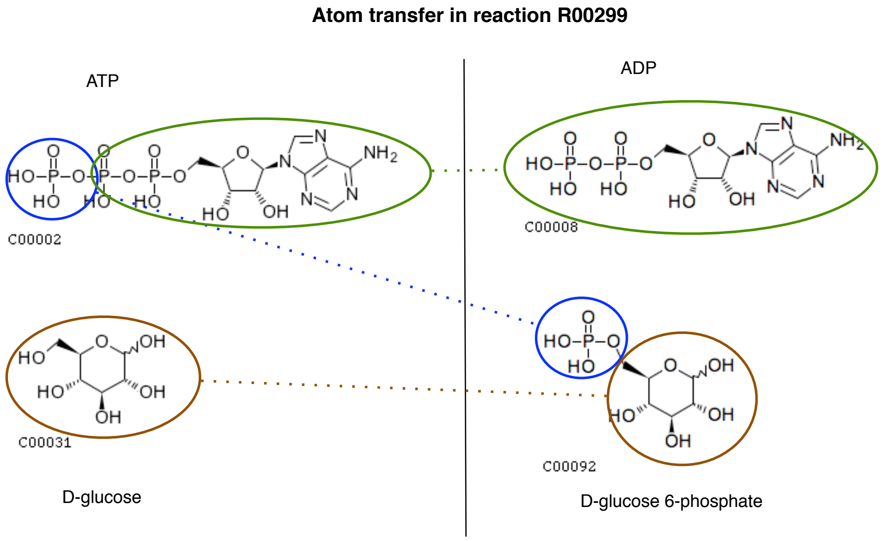


Figure 1.14: This Figure illustrates the transfer of atoms in a reaction that converts D-glucose and ATP into D-glucose 6-phosphate and ADP (KEGG identifier R00299). Corresponding atom groups on each side of the reaction are encircled with the same color. Atom transfers are represented by dashed lines.

defined as a sequence of biochemical reactions through which at least one carbon atom in X reaches Y" [7].

This definition allows to differentiate between main compounds carrying matter through a pathway and side compounds providing energy or electrons. Karp expresses a similar idea by defining main compounds as follows: "The main compounds lie along the backbone of the pathway - these compounds are shared between consecutive steps of a pathway" [88].

Karp's applies to all atom types, whereas Arita's is specific to carbon atoms. Thus, Arita's definition does not cover pathways proceeding via intermediates that do not contain carbon atoms, e.g. sulfide in the sulfur incorporation pathway.

When predicting metabolic pathways, an algorithm needs to know from which substrate of a reaction to go to which of its products. Applying Arita's definition helps to avoid side compounds such as ADP. Karp's definition is not helpful for pathway prediction, since it is pathway-specific. Thus, the pathway to be predicted needs to be known beforehand in order to classify compounds as main or side, which is an obvious contradiction. What is needed is a reaction-specific classification of compounds as main or side compounds. Reaction-specificity is required, since the role of a compound can differ from reaction to reaction. For instance, in the reaction with EC number 3.6.1.8 (conversion of AMP into ATP), ATP is a main compound, whereas it is a side compound in the example reaction presented in Figure 1.4. Such a classification of reactants has recently been presented by Kotera et al. for KEGG. Section 1.10.3 presents how metabolic pathway prediction can make use of this classification.

1.7.3 Feasibility-based definition of metabolic pathways

Recently, Esa Pitkänen has advanced a metabolic pathway definition that relies on the concept of feasibility [131]. Given a set of reactions R and a set of source compounds A, a *feasible metabolism* F from A is defined as the subset of R that includes all the reactions reachable from A. This means that a compound present in A can act either directly as a substrate of a reaction in F or indirectly as a substrate of a reaction in F after having been converted by other reactions in F. Thus, a reaction in F only involves compounds present in A or derived from compounds in A by other reactions in F.

A metabolic pathway is then defined as follows: "A metabolic pathway from A to [a target compound] t is any minimal feasible metabolism F from A to t, that is, removing any reaction from F leads to violation of requirement (i) [all reactions in F are reachable from A] or (ii) [t is among the products of the reactions in F]." (words in brackets added by Karoline Faust) [131].

To deal with side compounds, an *auxiliary compound set* S is defined. Compounds in this set are freely available as substrates without having to be produced from A. Whether a pathway is feasible or not therefore depends on the contents of A and S.

The main difference to the topological definition is that Pitkänen's definition only accepts pathways as valid that synthesize all the compounds contained in them, except those in the auxiliary set. Figure 1.15 shows a pathway that satisfies Pitkänen's definition.

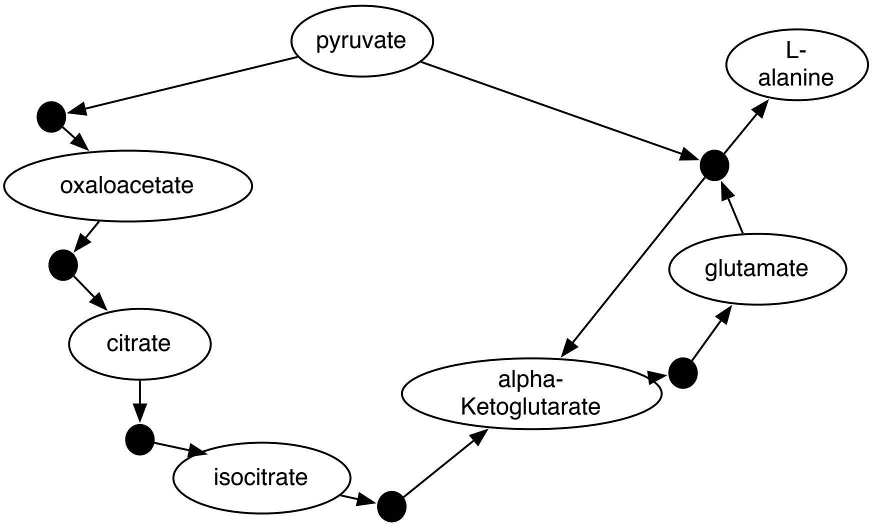


Figure 1.15: A feasible pathway from pyruvate to L-alanine. Pyruvate is a sufficient precursor to produce all intermediate compounds in this pathway, and no additional auxiliary compounds are needed. Adapted from Figure 1 in [131].

The scope of a compound as defined in [49] expresses an idea similar to the concept of feasible metabolism: The *scope of a compound* comprises all compounds that can be synthesized from it given a set of reactions.

1.7.4 Functional definition of metabolic pathways

In a personal communication, Jacques van Helden suggested a definition that emphasizes regulatory and functional aspects of metabolic pathways. He phrased it as follows: “Genes whose products are involved in a same metabolic pathways are generally (but not always) co-regulated. This regulation may differ from organism to organism. Different organisms may respond to the same metabolic requirement by expressing different sets of enzymes and transporters. The “boundaries” of a metabolic pathway should thus not be defined in terms of absolute rules, such as key compounds or stoichiometry, but be considered as organism- and even context-dependent. Other criteria can be used in addition to co-regulation, such as operons, synteny, horizontal gene transfer (e.g. in plasmids) or any other criterion revealing some functional association between sets of genes.”

1.7.5 Stoichiometry-based definitions of metabolic pathways

Stoichiometry-based definitions demand that a valid metabolic pathway stoichiometrically balances all its internal compounds. Compounds classified as external do not need to be balanced. The classification of compounds into internal and external is pathway-specific.

A famous stoichiometry-based definition of a metabolic pathway is the *elementary mode* (EM), defined as the "minimal set of enzymes that could operate at steady state with all irreversible reactions proceeding in the appropriate direction" [143]. *Extreme pathways* are similarly defined, but differ from elementary modes by their treatment of irreversible and reversible reactions. For the calculation of extreme pathways, each reversible reaction is split into two separate reactions for the forward and reverse directions, whereas in EM analysis, a number of constraints is placed on reaction directionality [125].

A special case of a stoichiometry-based definition is the *enzyme-subset* introduced in [129]. It defines metabolic pathways as enzyme subsets which are: "groups of enzymes that, in all steady states of the system, operate together in fixed flux proportions" [129]. These enzymes are considered to form linear metabolic pathways with the same steady-state flux and to be co-expressed simultaneously. This definition forms a link between the functional and stoichiometry-based definitions.

1.7.6 Metabolic pathway definition based on chemical organization theory

Chemical organization theory is a general concept that can be applied to any kind of network and which has applications in virus infection modeling [109], atmospheric photochemistry [24] and metabolism [23]. According to chemical organization theory, a metabolic pathway is considered as an organization if it is *self-maintaining* (all compounds can be re-generated by the pathway) and *closed* (all compounds that can be generated given the reactions of the pathway are part of the pathway).

The self-maintainance property combines stoichiometric balance with the idea of feasibility proposed in [131]. Both, self-maintainance and feasibility, require that all compounds in a pathway can be synthesized by the pathway. The self-maintainance property requires in addition that each pathway synthesizing a compound within the organization is stoichiometrically balanced.

Compounds that are not self-maintained can flow in or out of the organization. Importantly, compound concentrations can either remain constant or increase, which is an important difference to EM analysis and related methods (see section 1.10.3), which assume approximately constant compound concentrations.

Organizations can be ranked according to the number of different compounds they contain. A changing metabolic network may move up- or downward this hierarchy. Thus, chemical organization theory may be applied to describe the evolution of metabolic networks.

1.8 Which definition is most appropriate for pathway prediction?

Most of these pathway definitions have been conceived with specific applications in mind. For example, atom tracing experiments make use of the atom-flow based pathway definition. Thus, in my opinion, the appropriateness of a definition depends on the use one makes of pathway prediction. The functional definition for instance is most appropriate for the prediction of pathways from functionally associated genes, because it defines a pathway through gene association. When predicting pathways from sets of reactions or compounds, for instance to measure metabolic distances between enzymes [30], the classical definition (with some constraints to avoid biochemically meaningless pathways), might be more appropriate than the functional one.

1.9 Experimental validation of metabolic pathways

Different definitions emphasize different properties of metabolic pathways. Likewise, different experiments validate different aspects of metabolic pathways, such as the inability of an organism to synthesize a certain compound if the pathway is disrupted (validation of function) or the flow of atoms through the pathway (validation of flux).

Experiments can be performed at different levels, i.e. at the level of the pathway components (validation of interaction between enzymes and substrates, measurements of enzyme activity etc.), at the level of the pathway *in vitro* (in vitro reconstitution) or *in vivo* (mutagenesis, ^{13}C tracing, ...). For the reference pathways listed in the literature, the accumulation of evidence from different experiments over the years led to their widespread acceptance. When predicting metabolic pathways, it is important to be aware of the experimental techniques for their validation. Therefore, three selected whole-pathway validation techniques are briefly presented.

1.9.1 Mutagenesis

A mutant lacking an enzyme of the pathway may not be able to produce the end product of that pathway. If this disruption of the pathway renders the mutant unable to synthesize essential compounds, the mutant is said to be *auxotrophic* for these compounds, i.e. it can only grow when these compounds are supplied to it. Absence of the end product or accumulation of one of its predecessors supports the hypothesis that the enzyme is indeed involved in the pathway. A series of mutants, each lacking another enzyme of the pathway and accumulating another intermediate compound, form a strong evidence for the pathway in question.

This classical strategy of elucidating a metabolic pathway has been developed by the fathers of the one-gene-one-enzyme hypothesis, Beadle and Tatum. They identified two *Neurospora crassa* mutants, which were unable to produce tryptophan. The first mutant (strain 10575) could only grow in the presence of indole and accumulated anthranilic acid, which could be utilized for growth by the second mutant (strain 4008) [156]. From this and similar findings (e.g. [155]), it could be established that both anthranilic acid and indole are precursors of tryptophan and that indole is first synthesized from anthranilate and then converted into tryptophan (see also Figure 1.16).

A problem of mutant construction is the possible presence of isoenzymes or alternative pathways consuming the predecessor or producing the end product. In addition, a many-to-many relationship exists between genes and reactions (see section 1.3.3). Thus, a gene knock-out may not always produce a phenotype that can elucidate a pathway.

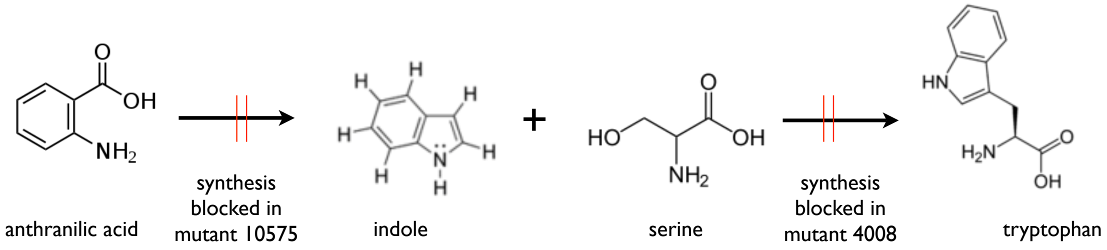

anthranilic acid
synthesis blocked in mutant 10575
indole
+
serine
synthesis blocked in mutant 4008
tryptophan

Figure 1.16: The steps of the tryptophan pathway that were elucidated by generating different *Neurospora crassa* mutants [156, 155].

1.9.2 Atom tracing

Atom tracing techniques (isotope labeling in combination with mass spectrometry or other techniques to identify labeled compounds) confirm that atoms flow indeed through the pathway in question (e.g. in [120, 172]).

1.9.3 In vitro reconstitution

Enzymes of a putative pathway are assembled together with the initial substrate and required cofactors *in vitro*. The formation of the end product is then confirmed with mass spectrometry or another compound identification technique (e.g. [173]).

1.10 Review on the computational prediction of metabolic pathways

In general, metabolic pathway prediction is the task of predicting biochemically valid metabolic pathways given a set of compounds or reactions of interest (referred to as the *seed* set) and metabolic data. More specifically, when given a group of associated enzyme-coding genes from an organism of interest, the task is to predict the specific metabolic pathway in which the gene products are involved. In addition to the metabolic network and the seeds, most pathway prediction approaches take further information into account (e.g. compound structures) in order to increase their prediction accuracy.

1.10.1 Pathway prediction and metabolic reconstruction

This thesis deals with *de novo* metabolic pathway prediction. In contrast, metabolic reconstruction relies in most cases on known pathways. *Metabolic reconstruction* aims at reconstructing the entire metabolism of an organism given its genome and additional information available in the literature or from databases. The reconstruction process can be carried out manually (e.g. [46]), automatically (e.g. [87, 114]) or semi-automatically (automated prediction combined with manual curation as in the second tier of BioCyc [22]). During automatic reconstruction, the enzymes of the organism of interest are mapped on a set of known pathways. This may result in gaps, if enzymes catalyzing reactions of the pathway are absent from the organism. Absence of enzymes may have the following reasons:

- The pathway is not present in the organism of interest. [124]
- A variant of the pathway is present in the organism of interest. [124]
- The missing enzymes have not yet been identified in the genome. [124]
- The reaction is spontaneous.

Sophisticated procedures have been developed to fill these gaps [92, 65], but they do not address the existence of alternative pathways.

However, an automated metabolic reconstruction procedure was published recently, which does not rely on pre-defined pathways and can therefore detect alternative pathways [28].

1.10.2 Prediction of metabolic pathways - the challenge

Metabolic pathways are highly interconnected, forming large complex networks as shown in Figure 1.17. Extracting relevant pathways from this "hairball" is a challenging task.

Pathway prediction algorithms have to deal with the exponential explosion of the number of possible pathways [111], because of the many reactions that form more than one product. As explained in section 1.7.1, not all topologically possible pathways are biochemically relevant. Thus, an algorithm has to sieve a large set of pathways to arrive at a subset of relevant pathways.

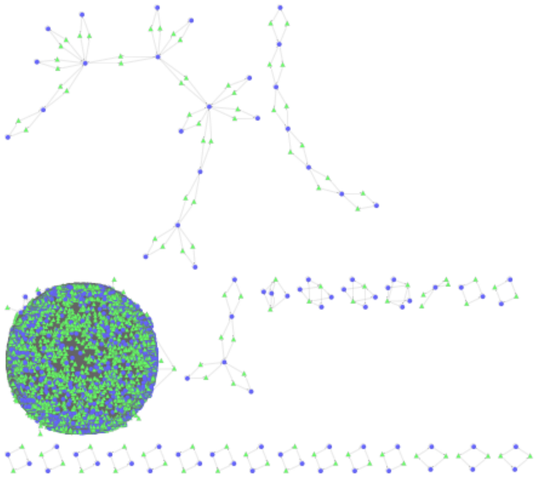


Figure 1.17: A bipartite, directed metabolic network constructed from KEGG (version 41.0) is shown. The network consists of several components, the largest of which is densely interconnected and contains the majority of compounds and reactions. The image was generated with Cytoscape [150].

The following constraints have been employed by various authors to restrict the number of possible metabolic pathways:

- Constraints on pathway length/weight
  - Minimal pathway length or weight: The result pathway should be as short (or as light) as possible (e.g. in [54, 31, 7, 138]). If no other objective is considered, this means that reactions or compounds can occur only once in the pathway and consequently cycles cannot be predicted.
  - Lower and upper boundary on pathway length or weight: The result pathway should not be longer, shorter or heavier than the indicated maximal or minimal length or maximal weight (e.g. in [31, 32]).
- Constraints on reaction directionality
  - Reaction directions: The pathway should be thermodynamically feasible, e.g. reactions should proceed in their physiological direction (e.g. in [129, 111]).
  - Mutual exclusion of reaction directions: The two directions of a reaction should not appear together in a result pathway (e.g. in [31, 111]).
- Constraints on pathway composition
  - Imposing or excluding compounds and/or reactions: Certain compounds and/or reactions should be present or absent in the result pathway (e.g. in [113, 18, 138, 111]).

- Constraints on compound production
  - Maximization of yields: The molar yield of a desired compound should be as high as possible [161].
  - Maximization of ATP production: The result pathway should produce a maximum of ATP [11].
- Constraints on compound stoichiometry
  - Stoichiometric balance: Internal compounds have to be stoichiometrically balanced (e.g. in [11, 129, 111]).
  - Minimal number of unbalanced intermediate compounds: The result pathway should minimize the number of unbalanced intermediate compounds [133].
- Constraint on pathway specificity
  - Optimization of *pathway specificity*: A compound is the more specific, the fewer the number of reactions is in which it acts as a substrate or product. The pathway specificity is the sum of the compound specificities in the result pathway and should be minimal [133]. The specificity concept was inspired by the observation that penalizing highly connected (i.e. unspecific) compounds improves pathway prediction [31, 32].

Metabolic pathway prediction approaches, as different as they may be, have two metabolism-specific problems to deal with:

- They have to treat hub compounds (also termed highly connected compounds or *high-presence compounds* by different authors). An algorithm that accepts hub compounds such as water, ADP or ATP as intermediates in a pathway will in most cases predict a biochemically wrong pathway (such as the one shown in Figure 1.13). However, hub compounds may be valid intermediates in some pathways (e.g. ATP in purine metabolism, see Figure 1.18).
- They have to deal with reaction directions. This problem is often solved by an appropriate graph representation (as discussed in 1.5). Alternatively, it can be tackled by introducing constraints on the reaction directions as in EM analysis.

If metabolic pathways are predicted from a set of query genes, genes have to be mapped to reactions, which is complicated by the many-to-many relationship between them. If the query genes are derived from a genome-scale experiment, scores associated to the genes have to be taken into account as well.

Metabolic pathway prediction from multiple seeds faces in addition the following challenges:

- The result pathway should not depend on the order in which seeds are provided.

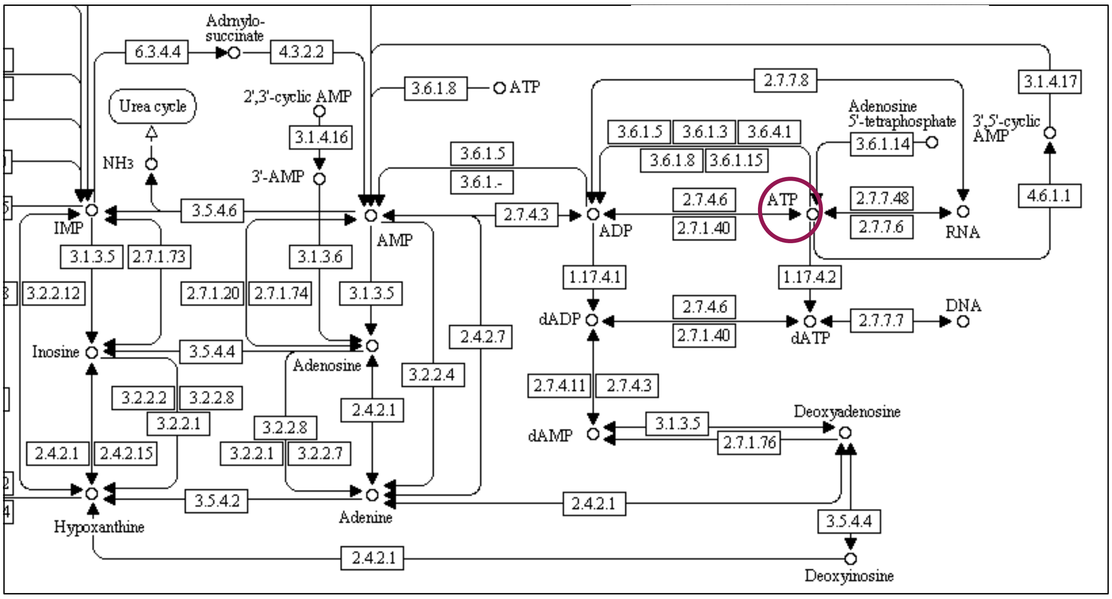


Figure 1.18: Zoom into the purine metabolism KEGG map. This Figure illustrates that the hub compound ATP is a valid intermediate in several pathways. ATP is encircled in maroon.

- It should be possible to provide groups of seed nodes. As mentioned in section 1.3.3, EC numbers are ambiguous and may comprise several reactions. A prediction approach that can handle seed node groups can treat all reactions of an EC number as belonging to the same group. As soon as one of the group members occurs in the predicted pathway, the seed node group is considered to be included in the pathway.

- It should be possible to integrate scores derived from high-througput experiments into the prediction approach in order to preferentially include reactions that are up- or down-regulated according to these data.

1.10.3 Two-end metabolic pathway prediction

Two-end metabolic pathway prediction is a special case of metabolic pathway prediction that predicts pathways from two seeds or seed sets. In case pathways are predicted from two seed sets, two-end pathway prediction is also called *multiple-to-multiple end pathway prediction*.

A remark on notation is needed here. Pathway prediction approaches are classified according to the number of seed nodes they take into *one-seed*, *two-seed* and *multiple-seed* approaches (synonymously called *one-end*, *two-end* and *multiple-end*). This thesis focusses on the two-seed and multiple-seed approaches.

Two-end metabolic pathway prediction has several applications, for instance:

- Metabolic reconstruction. Two-end pathway prediction can suggest organism-specific variants of metabolic pathways.

- Measurement of the metabolic distance between enzymes. Didier Croes measured the distance between enzymes that were associated in various ways (e.g. by protein-protein-interactions, fusion of their genes or grouping of their genes in operons) [30].

- Calculation of metabolic network properties. For example, Jeong et al. searched paths between all compound pairs in a metabolic network to measure its diameter [82] (see section 1.6).

The tables 1.4 and 1.5 summarize the major two-end metabolic pathway prediction approaches.

Table 1.4: Two-end metabolic pathway prediction strategies - First part.

Path finding strategy	Representation of metabolic knowledge	Treatment of hub compounds	Treatment of irreversible reactions	Number of reaction/ transformation rules currently treated by the approach	Implementation
Path finding	Metabolic knowledge is represented as a directed graph, often with compounds as nodes and reactions as arcs (other variants, such as bipartite graphs or reaction-centered graphs, exist).	Exclusion of a pre-defined list of highly connected compounds [56, 152, 164, 27]. Consideration of compound structure or atom tracing [6, 19, 112, 7, 138, 113]. Weighted graphs [31, 32]. Weighted graphs combined with compound structure [18, 54].	Graph structure (presence of direct and absence of reverse arc in a directed graph).	large (graphs may contain over 6,000 reactions)	Graph traversal (shortest paths or k-shortest paths algorithms)
Stoichiometric	Metabolic knowledge is represented as a set of linear equations (EM) or a set of constraints acting on sets of compounds and reactions.	Manual assignment [148, 111, 129, 11]. Automatic assignment [133]. Compounds assigned as external are not constrained to be balanced.	Specific constraints (e.g. EM analysis) or graph structure (e.g. extreme pathways analysis)	small (< 100 reactions) to intermediate (100 - 1,000 reactions)	Enumeration of solutions that satisfy the set of constraints or solve the set of linear equations. In early stoichiometric approaches, LISP is used to find solutions satisfying the given constraints [148, 111]. Sometimes, an objective function is formulated and a linear programming approach is adobted to solve the set of linear equations with respect to the objective function using available linear programming solvers such as CPLEX [11, 133].
Rule-based	Metabolic knowledge is represented as a set of automatically derived or expert-assigned transformation rules acting on a set of compounds.	Rules act on sub-structures (i.e. functional groups) of compounds.	Specific transformation rules.	intermediate (100 - 1,000 transformation rules)	Pathways are generated by iteratively applying the rule set on the query compounds (at the first step) or the transformed compounds of the nth generation (at the nth step). Combinatorial explosion is tackled by ranking rules or assigning a likelihood to them with the help of experts or machine learning approaches [80, 50]. Relative reasoning may further reduce the rule number [57]. Rule application is often carried out by a reasoning engine such as Prolog (MetabolExpert).

Table 1.5: Two-end metabolic pathway prediction strategies - Second part.

Possible constraints on pathways	Selected examples of computational validation	Selected examples of experimental validation	Availability of tools	Potential applications	Underlying pathway definition
Path length, path score (weight, compound similarity, etc.), absence/ presence of compounds/ reactions, reaction directionality	Accuracy of prediction measured for: 148 reference pathways from aMAZE and EcoCyc [32], 92 reference pathways from EcoCyc [18], 55 reference pathways from aMAZE [54]	^{13}C tracing in vivo [172] to validate pathway predicted by by ARM [6]	Several online tools available, see Table 1.6.	Prediction of drug biosynthesis [172] and waste compound biodegradation pathways [127]. Reconstruction of metabolic pathways from the genome [92].	Topological definition.
All topological constraints and additional stoichiometric constraint on internal compounds.	Accuracy of prediction measured for: 10 reference pathways from *E. coli* [11] 40 reference pathways from *E. coli* [133]	Design of mutant *E. coli* strains to optimize biomass production, EM predicted correctly production of biomass [161]	Several tools for EM analysis are available, e.g. YANA [146], METATOOL 5.0 [166] and CellNetAnalyzer [94]	Prediction of EMs for the design of knock-out/in strains with high yields for a compound of interest. Analysis of robustness of metabolic networks. Analysis of metabolic capabilities of an organism [162].	Stoichiometric definition.
Thresholds on rule probabilities.	Biodegradability prediction of perfluorinated compounds (CATABOL), drug metabolism predictions for 10 substrates (METEOR) [158], comparison of predictions for six unknown compounds by human experts and by prediction system (UM-PPS).	Predictions of the metabolism of three organonitrogen compounds by UM-PPS were consistent with enrichment cultures of microbes grown on these compounds as sole nitrogen source [75].	UM-PPS prediction system [75, 50] is publicly available, other pathway prediction tools are available as commercial software (e.g. METEOR, CATABOL, MetabolExpert).	Prediction of biodegradation of toxic compounds (UM-PPS, CATABOL), prediction of drug metabolism (METEOR, MetabolExpert).	Topological definition.

Path finding approaches

In this section, prediction approaches are presented that do not balance compounds and can therefore not predict stoichiometry. These approaches are also referred to as *path finding* approaches.

Path finding approaches rely in most cases on a graph traversal algorithm in order to enumerate paths between two compounds or reactions of interest. An exception is the path finding approach described in [42, 43], which relies on constraint programming. Path finding approaches are classified according to their treatment of highly connected compounds.

Tables 1.6 and 1.7 compare a number of path finding tools available on-line (i.e. with web interfaces). When compiling these tables, MetaRouter [127] and PathMiner [112] were not available at their published URLs (`http://pdg.cnb.uam.es/MetaRouter/` and `http://pathminer.uchsc.edu`, respectively), therefore they are not included.

Table 1.6: Path finding on-line tools - First part.

	Input options				Output options			
Tool name (reference)	Accepted seeds	Multiple-to-multiple seed path finding	Organism-specific network construction from KEGG PATHWAY	Obligatory absence or presence of user-provided compounds and/or reactions in predicted paths	Paths filtering options	Paths ranking criterion	Display options	Metabolic source databases
FMM [27]	Compounds	No	No	No	None	No criterion defined	None	KEGG PATHWAY/ LIGAND, UniProtKB/ Swiss-Prot, dbPTM
Rahnuma [113]	Compounds	Yes	Yes, various options to construct networks specific to a set of organisms or a given phylogeny	A user-provided or a pre-defined list of compounds can be excluded. A list of elements that have to be present in the paths can be given.	Path length	Path length and connectivity score (all path of the same length are ranked according to this score)	HTML or simple text	KEGG PATHWAY/ RPAIR and manually compiled atom mappings for carbon and nitrogen in some KEGG maps.
Metabolic Pathfinder [54]	Compounds, Reactions, Reactant pairs, EC numbers.	Yes	Yes, via KEGG network provider. Networks can be RPAIR networks.	Yes	Maximal rank index of paths, maximal path weight, minimal path length, maximal path length.	Path weight (result table can be sorted according to path rank, path weight or path length)	Output format (table or separate paths graphs or paths unified into one graph). Graph format (dot, gml, visml, tab-delimited). Results sent by email or displayed in browser.	KEGG LIGAND/ RPAIR or optionally KEGG PATHWAY
MetaRoute [18]	Compounds	No, but one-to-all path finding supported.	Yes	Yes, (in subsequent steps).	Maximal paths number	Compound or reaction weight, path length	None	KEGG PATHWAY/ LIGAND, atom mapping rules computed from KEGG and EcoCyc data.
PathFinder [31, 32]	Compounds, Reactions, EC numbers.	No	No	No	Number of ranks, maximal path length, maximal path weight.	Path weight	Image format (svg, png)	KEGG LIGAND
PHT (Pathway Hunter Tool) [138]	Compounds	No	Yes	Compounds to be present as well as EC numbers to be excluded can be specified.	Minimal path length	Path length and compound similarity	None	KEGG PATHWAY, BRENDA, PROSITE
ARM (Atomic reconstruction of Metabolism) [6, 7]	Compounds	No	No	Yes (exclusion only). Reactions can be specified by their name or EC number, compounds by their KEGG identifier or name.	None	Path length	Atom tracing versus drawing mode	KEGG LIGAND, EcoCyc and Roche Pathway chart, data curated and atom mappings computed by the author.

Table 1.7: Path finding on-line tools - Second part.

Path finding strategy				Output	Evaluation	Strengths	Weaknesses	
Presentation of the metabolic data	Treatment of reaction directionality	Algorithm	Treatment of hub compounds					Tool name URL
Directed compound graph (nodes: compounds, arcs: reactions)	Directionality presented as given in KEGG PATHWAY (reversible reaction is represented by forward and reverse arc).	Breadth first search with subsequent ranking of paths according to the number of crossed KEGG maps.	Exclusion of a pre-defined compound list	Unsorted list of paths, each path is displayed with its image and link to KEGG maps crossed by it. List of organism-specific enzymes participating in predicted paths.	None	Well drawn pathway images. Easy-to-use tool interface.	Knowledge of KEGG pathway maps required. Paths containing excluded compounds cannot be inferred, e.g. pyrimidine de novo biosynthesis from carbamoyl-phosphate.	FMM http://fmm.mbc.nctu.edu.tw/
Directed hypergraph (nodes: compounds, hyperarcs: reactions)	Directionality presented as given in KEGG PATHWAY (reversible reaction is represented by forward and reverse hyperarc).	Backtracking (Depth first search)	Manually compiled atom mappings or KEGG RPAIR. Optionally exclusion of a pre-defined compound list.	List of paths, with compounds linked to KEGG compound entries (in HTML mode).	None	Tracing of different atom types. Distinction between curated and automatically generated KEGG maps. Many options to construct a network from sets of organisms in KEGG PATHWAY. Can determine reactions whose removal will destroy all paths between selected compounds. Comparative analysis of organism-specific networks and pathways.	No images of paths generated. Paths are not available as a network. Choice of seed compounds is tedious. Organism-specific networks cannot be downloaded.	Rahnuma http://portal.stats.ox.ac.uk:8080/rahnuma/
Bipartite directed graph for KEGG LIGAND and bipartite undirected graph for KEGG RPAIR with separate node sets for reactions and compounds	All reactions treated as reversible, but user may provide or construct custom networks where given directionality of reactions is kept.	REA k-shortest paths algorithm [83]	KEGG RPAIR, weights.	Either sortable pathway table with entries linked to KEGG or graph. Graph image generated. Export of graph into selected format.	55 reference pathways annotated in aMAZE [101] from three organisms (*E. coli*, *S. cerevisiae* and *H. sapiens*)	Reactions and EC numbers as input supported. Custom networks can be supplied in various formats (tab-delimited, gml and KGML). Batch job processing supported. Tool is available on command line and as a web service for easy combination with other tools. Local installation of web server possible. Parsers are distributed with web server, so users can update data used by the tool. Integration into NeAT allows submission of paths to subsequent visualization or analysis steps. Several pre-defined weighting schemes are offered, custom weights can be submitted.	Very simple graph layout. Tracing of different atom types not possible.	Metabolic Pathfinder http://rsat.ulb.ac.be/neat/
Directed reaction graph (nodes: reactions, arcs: compounds)	All reactions treated as reversible, but optionally irreversible reactions can be specified.	Eppstein's k-shortest paths algorithm [51]	Atom tracing, weights.	List of pathways, each as separate graph and unified into one graph. Graph image created for each of these. Export into SBML and METATOOL input format.	137 reference pathways from EcoCyc	Very good graph layout of predicted paths, graph image includes side compounds. Tracing of different atom types possible (carbon, nitrogen, sulfur and phosphor). Consideration of compound hierarchy relations. Filtering of unbalanced reactions and compounds without structural information. Comparison of pathways between two organism sets is possible. Custom networks can be supplied.	Data outdated (last update in 2007).	MetaRoute http://www-bs2.informatik.uni-tuebingen.de/services/MetaRoute/
Bipartite	All reactions treated	Backtracking	Weighted graph,	Pathway list and	56 aMAZE	First path finding	Construction of	PathFinding

directed graph with separate node sets for compounds and reactions	as reversible.	(Depth first search)	optionally an unweighted or filtered graph (pre-defined compounds excluded) can be selected.	image is sent by email.	pathways from *E. coli* and *S. cerevisiae*, 92 pathways from EcoCyc	tool that has been carefully evaluated. Compound structures or atom mappings not required. Easy-to-use tool interface.	organism-specific KEGG PATHWAY networks or submission of custom networks is not possible.	http://www.scmbb.ulb.ac.be/Users/didier/pathfinding/metabpathfinding.php
Directed compound graph (nodes: compounds, arcs: reactions)	Directionality presented as given in KEGG PATHWAY (reversible reaction is represented by forward and reverse arc).	Breadth First Search with constraints (chemical similarity between compounds)	Global and local compound similarity	List of paths, each with compound structure and links to KEGG entries. Paths can be downloaded in gml or stoichiometric matrix format. Organism information displayed. Matrices with reaction/enzyme occurrences in selected organisms available.	2 pathways	A list of EC numbers can be provided to construct a custom network. Pathway output includes a table displaying which reactions occur in which of the selected organisms. Analysis and download of organism-specific networks is possible. Hub compounds and reactions as well as load and choke points are listed.	Result page is difficult to read. Data is outdated (last update in 2006)	PHT http://pht.tu-bs.de/PHT/
Directed compound graph (nodes: compounds, arcs: reactions)	Directionality presented as given in KEGG LIGAND. Directionality of reactions can be modified.	Eppstein's k-shortest paths algorithm	Atom tracing	List of paths, each path associated with an image showing the compound structures. More paths can be added to the list with a mouse click.	Correct tracing of atoms has been systematically validated, but accuracy of recovery of known pathways was not measured.	Well drawn pathway images including compound structures. Tracing of different atom types: carbon, nitrogen and sulfur are supported. Visualization of atom tracing: Positions of selected atom in path compounds is highlighted on mouse click. Pathways can be modified by the user with a chemical structure drawing tool. Applet visualizes structures of substrates and products, which can be searched by entering compound or reaction KEGG identifiers or names, reaction equations or EC numbers, manually drawn compound structures or compound formulas.	User interface is not intuitive. Java applet is inconvenient to close (browser has to be closed). Export of paths not possible.	ARM http://www.metabolome.jp/software/plonejavaapplet.2006-06-29.9364604635/

A. Path finding without hub compound treatment

As mentioned in 1.6, a "small world" property of metabolic networks is postulated in [82], i.e. in most metabolic networks, any two compounds can be connected by a short path (of length around three). However, these authors failed to treat highly connected compounds and therefore based their analysis on biochemically invalid pathways. Several authors have repeated this analysis and have demonstrated that the average shortest path length is much higher if hub compounds are treated appropriately. For instance, the average shortest path length between compounds in the *Escherichia coli* network was measured independently in [107] and [7] to be around eight, where the former authors treated hub compounds by excluding them and the latter avoided them by atom tracing. The distribution of average path lengths between reactions in the complete KEGG network peaked between five and eight reaction steps [32].

Currently, no path finding tool exists that does not treat hub compounds.

B. Path finding with exclusion of highly connected compounds

Many authors treat highly connected compounds by excluding them from the network [56, 164, 152, 107, 27]. This strategy has the drawback that pathways containing those compounds as intermediates cannot be predicted. For instance, ADP and ATP are often excluded as typical hub compounds, but are valid intermediates in the purine biosynthesis pathway. In addition, the distinction of a non-hub from a hub node requires additional parameters (e.g. a threshold on the node degree).

FMM [27] is an example for a path finding tool that treats hub compounds by excluding them. Consequently, it fails to find a path between carbamoyl-phosphate and CTP (the start and end point of pyrimidine ribonucleotides de novo biosynthesis). Remarkably, this path finding tool is the only one in the tables 1.6 and 1.7 that predicts the glycolysis pathway correctly from glucose and pyruvate. The reason is that pathways are post-filtered with KEGG pathway maps. Strictly speaking, because of its use of KEGG maps, FMM is not a de novo pathway prediction tool as the others discussed in this section. It has been included in tables 1.6 and 1.7 because it shares many properties with other path finding tools.

C. Path finding in weighted networks

In [31, 32], Didier Croes presented a path finding approach that avoids highly connected compounds by assigning weights to the nodes of the metabolic network. The weights are assigned such that sparsely connected compounds are preferentially traversed by the path finding algorithm. More precisely, each compound receives a weight equivalent to its degree, that is to its number of incoming and outgoing arcs. Reactions receive a weight of one. The backtracking algorithm developed by Fabian Couche [29] enumerates the lightest paths in the weighted network. Path finding in weighted networks can identify relevant metabolic pathways without the need of compound exclusion or compound structures.

In addition, Didier Croes compared the performance of the weighted network to a filtered network (where 36 highly connected compounds were removed) and to an untreated "raw" network. The evaluation was carried out on metabolic networks constructed from KEGG and EcoCyc [90] (which is a part of BioCyc) and proceeded as follows: First, a pathway was predicted given the start and end reaction of a linearized reference pathway. Then, the path finding accuracy was computed by comparing the node sets of the predicted and the reference pathway. This procedure was repeated on 56 pathways from aMAZE [101] and 104 pathways from EcoCyc. For both, KEGG and EcoCyc, the weighted network clearly outperformed the filtered and the raw network.

A path finding tool accompanied the evaluation (PathFinding, [31]). This tool is no longer available, but is listed in tables 1.6 and 1.7 because it is the predecessor of the Metabolic Pathfinder tool developed during this thesis.

Didier Croes' work was the starting point for this thesis, which owes not only the use of a weight policy to him but also ideas on path finding accuracy measurement and metabolic network construction. In addition, Didier Croes inspired the improved path finding approach presented in [54].

D. Path finding considering compound structure

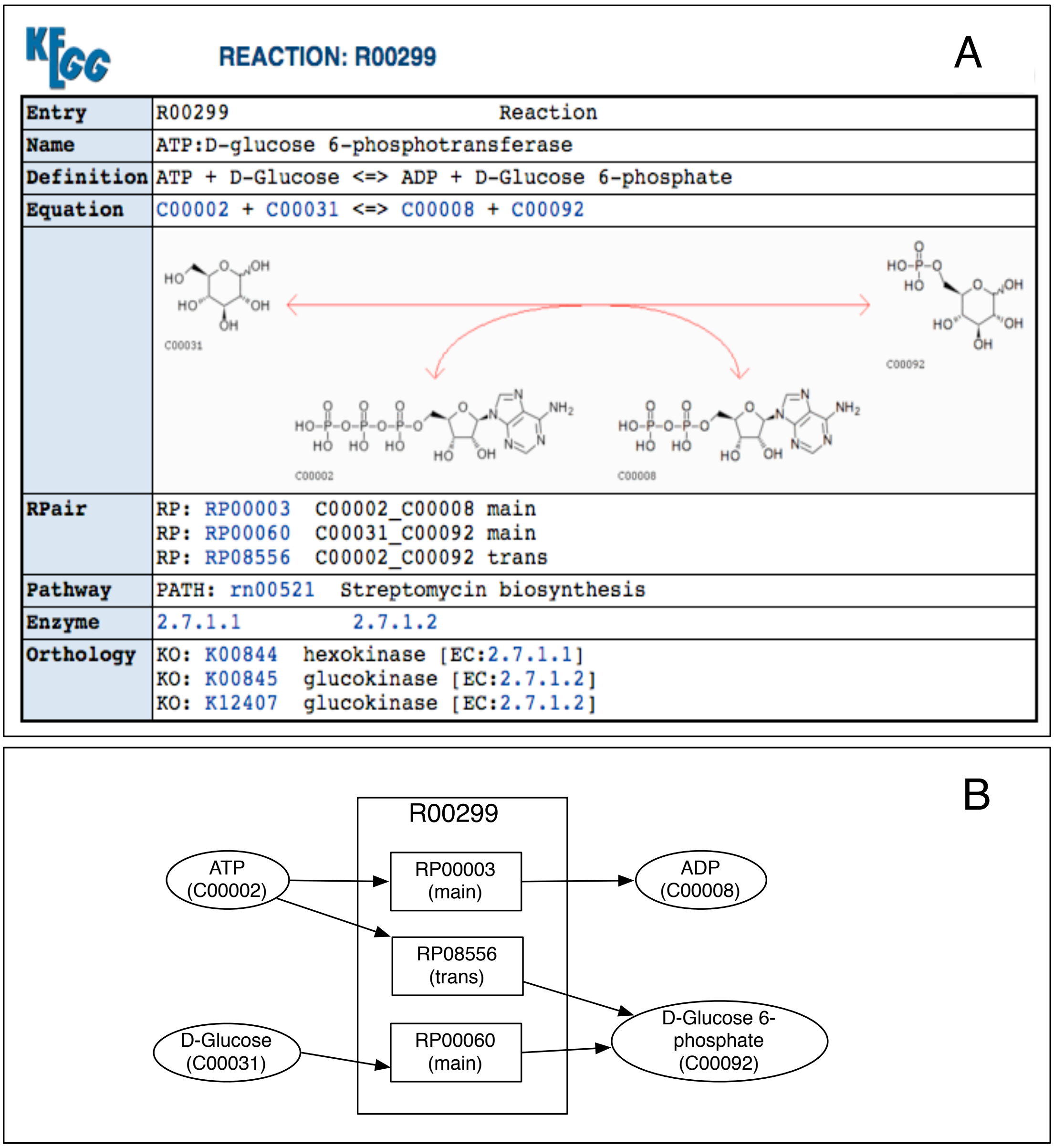
Faced with the hub compound problem, many authors developed different solutions that all relied on the idea of taking the structure of compounds into account.

As mentioned in 1.7.2, Arita introduced the idea to trace atoms through metabolic networks in silico [5] and developed a path finding tool (ARM, atomic reconstruction of metabolism) based on this idea [6, 7]. *Atom tracing* proceeds as follows: For each reaction in the metabolic network, each of its substrates is paired with each of its products. For each substrate-product pair, the structures are mapped such that corresponding atom positions can be identified. To do so computationally means to solve the maximum common subgraph problem, which is NP-hard. Arita developed an efficient heuristic, which minimizes breakage and formation of chemical bonds [5, 6]. In [19], this heuristic is combined with maximal partial injections, which allow to find reactions that transfer a maximal number of atoms. Other authors do not trace atoms, but instead compute overall compound similarities. The compound structure is first reduced to a descriptor, which is either a vector in a chemical state space [112] or a string generated with dedicated software [138]. In a second step, a distance measure between these descriptors is introduced, which allows to search for paths that maximize compound similarity.

Atom tracing considers the two-dimensional structure of compounds, whereas compound structure descriptors simplify compound structure further to one dimension, allowing for quicker search and comparison. In the absence of a comparative evaluation of tools based on two-dimensional compound structure (such as ARM) or one-dimensional compound structure description (such as Pathway Hunter Tool [138]), it is not clear whether a simplification from two to one dimension means a loss of path finding accuracy.

With the RPAIR database, manually compiled reactant pair mappings became available [97, 96]. Each of these reactant pairs has a role assigned to it, such as *main* ("main changes on substrates" [96]) or *trans* ("focused on transferred groups for transferases" [96]). Table 1.8 lists all such roles as defined in [96].

With these reactant pairs at hand, it is possible to identify the main and side compounds of a reaction. For instance, a naive algorithm could cross the reaction shown in Figure 1.4 from D-glucose to ADP or from ATP to D-glucose-6-phosphate. An informed algorithm taking into account the RPAIR annotation will avoid these irrelevant paths, because the pair ATP/D-glucose-6-phosphate has the role *trans* instead of *main* and the pair D-glucose/ADP does not even exist (since no atoms are transfered between these two compounds). Figure 1.19 gives an illustration of this example.



REACTION: R00299

Entry	R00299 Reaction
Name	ATP:D-glucose 6-phosphotransferase
Definition	ATP + D-Glucose <=> ADP + D-Glucose 6-phosphate
Equation	C00002 + C00031 <=> C00008 + C00092
RPair	RP: RP00003 C00002_C00008 main RP: RP00060 C00031_C00092 main RP: RP08556 C00002_C00092 trans
Pathway	PATH: rn00521 Streptomycin biosynthesis
Enzyme	2.7.1.1 2.7.1.2
Orthology	KO: K00844 hexokinase [EC:2.7.1.1] KO: K00845 glucokinase [EC:2.7.1.2] KO: K12407 glucokinase [EC:2.7.1.2]

Figure 1.19: Figure A shows the reactant pair decomposition of KEGG reaction R00299, whereas Figure B depicts more clearly the relationships between the reactant pairs and their substrates and products. The reaction can be decomposed into three reaction pairs, two main reactant pairs reflecting the transfer of large atom groups (ATP/ADP and D-glucose/D-glucose 6-phosphate) and one trans reactant pair (ATP/D-glucose 6-phosphate), which describes the transfer of the phosphate group. These three reactant pairs correspond to the three atom groups shown in Figure 1.14 for the same reaction. Importantly, there is no reactant pair between D-glucose and ADP, since no atoms are transferred between these two compounds. Note also that each reactant pair has only one substrate and one product.

Table 1.8: Roles of reactant pairs as defined in [96]

Reactant pair role	Definition of role
main	main changes on substrates
trans	changes on cofactors for oxidoreductases
cofac	transferred groups for transferases
ligase	consumption of nucleoside triphosphates for ligases
leave	separation or addition of inorganic compounds for such enzymes as lyases and hydrolases

As a consequence of the reaction-specific separation between main and side compounds, the KEGG RPAIR network features top ten hub compounds that differ from those of the KEGG LIGAND network (see Table 1.2). For instance, the typical side compounds NAD(P)$^+$/NAD(P)H disappear from the KEGG RPAIR hub compound list, because the number of reactant pairs involving them is much smaller than the number of reactions involving them. In contrast to KEGG LIGAND and MetaCyc, highly connected compounds of the central metabolism (e.g. pyruvate and glutamate) appear on the KEGG RPAIR list. They are not listed for KEGG LIGAND and MetaCyc, because they are less connected than the side compounds.

The path finding tool Rahnuma makes use of the RPAIR annotations [113]. ARM [6] is an example for a path finding tool that traces atoms. Moreover, ARM is currently the only path finding on-line tool that visualizes atom tracings.

E. Path finding in weighted networks integrating compound structure

Recently, two approaches have been published that combine weighted networks with compound structures [54, 18]. Both make use of the weight policies introduced in [31, 32].

During this thesis one of these approaches was developed, which is presented in detail in chapter 2. Briefly, the approach consists in the construction of weighted KEGG RPAIR networks, which are compared to KEGG LIGAND networks and unweighted KEGG RPAIR networks [54]. In [18], a new procedure to compute atom mappings is presented, which is based on a minimum cut algorithm acting on SMILES. *SMILES* (Simplified Molecular Input Line Entry System) are strings that describe compound structure, including stereoisomers [168] (see Figure 1.3). If several atom mappings are possible for a reaction, the one that occurs most frequently in the reaction's EC cluster is selected. The *EC cluster* is the set of all EC numbers sharing the first three digits with the reaction's EC number and thus its mechanism. From the atom mappings, a directed network is constructed where each node represents a substrate-product pair (equivalent to a reactant pair of KEGG RPAIR) and where each arc represents a compound shared by two substrate-product pairs. The performance of this network under various weight policies was assessed.

For both approaches, thorough evaluation showed that a weighted network penalizing hub compounds and integrating compound structure reached the highest path finding accuracy.

Both approaches are available as on-line tools: Metabolic Pathfinder [54] (http://rsat.ulb.ac.be/neat/) and MetaRoute [17] (http://www-bs2.informatik.uni-tuebingen.de/services/MetaRoute/). Metabolic Pathfinder is currently the only path finding on-line tool that accepts reactions as source and target, whereas MetaRoute is the only path finding on-line tool that offers tracing of four atom types.

Stoichiometric pathway prediction

Stoichiometric pathway prediction constraints the solution space further than path finding by only accepting stoichiometrically balanced paths. Thus, in contrast to path finding, this pathway prediction approach can also predict stoichiometric coefficients.

A. Early contributions

In one of the first articles on computational metabolic pathway prediction, a search algorithm written in LISP acts on a database including 100 compounds and 70 enzymes [148]. For each compound, the elements it carries are listed (e.g. carbon, nitrogen, . . .), which facilitate balances on carbon, phosphorus, sulfur, and nitrogen. Energy carriers such as ATP and electron donors such as NADPH are freely available, all other compounds involved in a pathway have to be synthesized by it. A pathway can be predicted by defining an "initial state" (equivalent to a source compound) and by applying enzyme "operators" on it until a a "target state" (a target compound) has been reached. An enzyme and its reaction are represented as a single entity, which is annotated with the EC number, the physiological reaction direction, the reactants with their stoichiometric coefficients, the enzyme activators, inhibitors and known pathways. The system was tested by providing glucose-6-phosphate and pyruvate as seeds, which yielded the Embden-Meyerhof-Parnas (that is glycolysis), Entner-Doudoroff, and pentose phosphate pathways.

Some years later, Mavrovouniotis [111] extended this work by introducing new constraints to the system, which still consists of a metabolic database (with 250 reactions and 400 compounds) and a search algorithm written in LISP. The constraints include those on the thermodynamic feasibility of a reaction, i.e. only reactions are allowed that can proceed in the predicted direction under physiological conditions. As described in section 1.3.2, the direction of a reaction depends on its Gibbs free energy change. Mavrovouniotis developed a group contribution method to calculate this quantity. Other constraints forbid or require selected compounds and reactions to occur in predicted pathways and limit the pathway size. Hub compounds are treated as in [148] as freely available compounds, which do not need to be balanced or synthesized. Mavrovouniotis analyzed the production of lysine from glucose and ammonia as a case study. He identified kinetic bottlenecks given reactant concentrations and enzyme rates and suggested alternative pathways to circumvent these bottlenecks. He also introduced the lumping of a linear sequence of reactions into one reaction and the ranking of pathways by yield.

B. Elementary mode analysis

ELementary mode analysis does not take two seed nodes and can therefore not be considered as a two-end pathway prediction technique. But since it has been compared with path finding [38, 132], it will be treated here.

The definition of *elementary modes* (EM) is given in section 1.7.5. In the following, the computation of EMs is summarized (see e.g. [129, 162]).

The development of compound concentrations over time can be described by a set of linear differential equations involving the reaction rates and the stoichiometry of compounds participating in a reaction. These equations can be written in a compact form using matrices and vectors:

$$\frac{dC}{dt} = S * v \tag{1.2}$$

where v is the vector of *reaction rates* (also called *fluxes*), S the *stoichiometric matrix* and C the *vector of compound concentrations*. Each row of the stoichiometric matrix represents a compound and each column a reaction, whereas each entry is the stoichiometric coefficient of the i*th* compound in the j*th* reaction.

The basic assumption of EM analysis (as well as *flux balance analysis* and *metabolic flux analysis*) is that due to the fast turn-over of compounds, metabolism can be considered to be at steady state (chemostat) or at pseudo steady state (batch culture) at relevant time scales. Thus, compound concentrations do not change over time and consequently the left-hand side of equation 1.2 can be set to zero:

$$S * v = 0 \tag{1.3}$$

With this simplification it is possible to compute the solutions that satisfy equation 1.3 and some additional constraints concerning reaction directionality. Geometrically, the solution space forms a convex polyhedral cone. Each solution is a vector of fluxes. EMs are special solution vectors satisfying an additional *non-decomposability constraint*, which roughly states that no reaction can be removed from an EM without disturbing it as a functional unit [125]. EMs can be computed with tools such as METATOOL [166] or efmtool [157].

In EM analysis, compounds are classified into *internal compounds*, which need to be balanced and *external compounds*, which are assumed to be freely available and are therefore not balanced. Likewise, reactions are classified into *internal reactions*, which take place inside the system and *exchange reactions*, which cross system borders (e.g. transport). Metabolic networks need to meet certain requirements in order to apply EM analysis on them: for each reaction, all concerned elements (i.e. atoms and ions) and their charges need to be balanced. Parallel pathways, null cycles, dead ends or erroneous exchange reactions need to be identified and removed. In order to simplify the metabolic network, linear sequences of reactions are lumped into one reaction, e.g. the last four steps of glycolysis in [38]. Metabolic networks are usually small and carefully constructed by hand. However, recent advances in EM computation allow their application to genome-scale metabolic networks [85, 37, 157].

De Figueiredo et al. compared EM analysis to path finding tools (namely PathFinder and Pathway Hunter Tool) [38]. In [53], we pointed out weaknesses of this comparison and in addition enumerate advantages and disadvantages of EM analysis and path finding for metabolic pathway prediction. This discussion can be found in chapter 5. A comparison of both approaches has also been published in [132].

The solution space of equation 1.3 is spanned by flux vectors called extreme pathways (e.g. [125]). Since the *extreme pathways* were apparently not yet employed for metabolic pathway prediction, they will not be further discussed here.

C. Other stoichiometric approaches

In [11], a stoichiometric approach is presented that combines constraints with an *objective function*. Thus, in contrast to EM analysis, the solution space is narrowed to one solution, which optimizes the objective function. The objective function is a mixture of two objectives: (1) minimize the number of reactions in a pathway and (2) maximize the ATP production.

Compounds are classified as *low presence* and *high presence* compounds depending on the number of reactions they are involved in. As the external compounds in EM analysis, "high presence" compounds do not need to be balanced.

Constraints include those for low presence compound balancing and for mutual exclusion of the two directions of one reaction. In particular, a number of constraints concerns cycles; e.g. cycles involving more than one high presence compound are forbidden.

With these objective function and constraints, a number of known pathways could be recovered from the *Escherichia coli* metabolic network, among others glycolysis.

In a second article [133], this constraint-based pathway prediction is further refined:

- Prediction of more than one paths (K paths) between a source and a sink allows to deal with branched pathways.

- "Low presence" and "high presence" compounds are no longer manually assigned beforehand, but automatically during pathway prediction.

- Another optimization function is introduced, which minimizes the number of unbalanced main compounds and maximizes the specificity of the pathway. The pathway specificity depends on its number of specific compounds. A compound is the more specific the less it is involved in other reactions.

- Additional constraints for the treatment of highly connected compounds are introduced, e.g. inorganic compounds or high presence compounds are not allowed to occur in a path if source and sink can be connected by other compounds.

Pathway prediction has been evaluated on 40 reference pathways from *Escherichia coli*, 36 of which were exactly recovered [133].

Rule-based pathway prediction

Rule-based pathway prediction consists in the iterative application of transition rules on a set of compounds. It may proceed as one-end prediction (e.g. to propose a number of different degradation pathways for one query compound) or two-end prediction (after a desired target compound was obtained, the iterative application of rules is stopped). The rules are defined such that hub compounds are avoided.

Rules describe generalized reactions, which cover a set of specific reactions. Usually, a rule applies to a certain sub-structure (e.g. functional group) of a compound class, e.g. the rule "primary Alcohol converted to Aldehyde" (rule identifier bt0001 in UM-BBD [50]) is applicable to all compounds with an alcoholic group.

Rules are often manually annotated by experts (e.g. in UM-PPS [50], METEOR [66] or MetabolExpert [35]) or automatically derived by generalizing reactions (e.g. [80, 69]).

In order to deal with combinatorial explosion, rules are ranked with the help of expert knowledge (e.g. [50]) or by machine learning approaches (e.g. [57]).

A number of rule-based tools is available, which are specialized on different metabolic networks. For example, METEOR and MetabolExpert are commercial expert systems tailored to drug metabolism, whereas CATABOL (commercial) and UM-PPS (freely available) are designed to predict biodegradation pathways.

In contrast to the path finding and stoichiometric approaches, the metabolic fate of compounds not yet present in a database can be predicted, since rules apply to compound sub-structures and not to the whole compound. Most rule-based pathway prediction tools are commercialized, probably because carefully designed rules allow them to reach a high positive predictive value. However, without a systematic evaluation, it is unclear whether these tools also reach a high accuracy, because overly restrictive rules might decrease their sensitivity.

1.10.4 Multiple-end metabolic pathway prediction

In the previous section, metabolic pathway discovery with two seeds or seed sets was presented, where seeds act as source and target compound(s) and/or reaction(s) respectively. A more ambitious goal is the prediction of a metabolic pathway from a seed group (or several seed groups). For this, an algorithm is needed that extracts a relevant sub-network from the input network given the seed nodes. Metabolic pathway prediction from several seeds has received much less attention than prediction given two seeds. Therefore, sub-network extraction techniques applied to other biological networks will be discussed here as well.

There is a clear distinction between the *inference of a network* from microarray or other data sets (as in [171]) and the extraction of a sub-network from a network. The goal of the former approach is to predict (at least partially) a network from high-throughput data (e.g. regulatory networks from gene expression profiles), whereas the goal of the latter approach is to predict a pathway from a network that is already known. In sub-network extraction, the network can be weighted with high-througput data, but these weights do not affect the topology of the network itself.

The development of multiple-seed pathway extraction techniques has been driven by the need to interpret high-troughput data sets. Zien et al. [177] were among the first to advance a technique that instead of mapping genes on pre-defined pathways enumerated possible pathways in a metabolic network and selected the pathway that fitted best the given gene expression data. However, their work was restricted to two-end pathway prediction.

Tables 1.9 and 1.10 summarize a selection of different multiple-seed sub-network extraction approaches developed during the last years. These approaches may be classified into global and local approaches according to seed node treatment. *Global approaches* ([77, 139, 119]) select nodes in the network during execution of the algorithm such that the resulting sub-network optimizes a criterium (usually sub-network weight) whereas *local approaches* receive a set of seed nodes beforehand which are then connected ([147, 40, 3, 4]). However, seed-node specific sub-networks can also be extracted with global approaches, by assigning appropriate weights to the network nodes (low weights to the seeds, high weights to all other nodes).

Sub-network extraction has been applied to protein-protein and protein-DNA interaction networks [77, 147, 40], metabolic networks [119, 3, 4] and networks integrating both of the former [139].

Usually, sub-networks are extracted from weighted networks. Often, the weights represent scores derived from one or several high-throughput experiments (microarray data in [77, 40] or enzyme levels in [119]). This thesis focusses on sub-network extraction and does not make use of high-throughput data to weight the networks. However, since the weight assignment step is clearly separated from the sub-network extraction step, the procedures applied in this thesis can be easily combined with more sophisticated weight policies.

Many authors developed their own heuristics to solve the sub-network extraction problem [77, 139, 119, 3, 4]. Others extract sub-networks with *Steiner tree algorithms* [147, 40]. As described in section 9.1, a Steiner tree algorithm searches for the sub-network that connects the given seed nodes in the given network with minimal weight. This problem is known to be *NP-hard*, which means, roughly stated, that it cannot be solved in polynomial time. Nevertheless, some Steiner tree algorithms have been published that solve the problem exactly (though not in polynomial time). For instance, the Dreyfus-Wagner algorithm [45] is applied with modifications in [147] and Ljubić's exact solution [105] is applied in [40].

Most sub-network extraction techniques are not accompanied by web servers or ready-to-use applications, with the exception of [77] (Cytoscape plugin) and [3] (two web servers). In addition, only few approaches have been evaluated ([139, 40]). Thus, the evaluation presented in [55] and the pathway prediction web server developed during this thesis are useful contributions to the prediction of metabolic pathways from multiple seeds.

Table 1.9: Multiple-end pathway prediction strategies - First part.

Subgraph extraction strategy (reference)	Biological network	Omics data mapped on network	Network weight assignment	Treatment of hub nodes	Metabolic networks only: treatment of reaction directionality
Heuristic based on simulated annealing (Ideker et al. [77])	Undirected protein-protein and protein-DNA interaction network (small network: 362 edges, obtained from GAL pathway perturbation study in yeast [78], large network: 7,462 edges from BIND [8] and TRANSFAC [110])	Gene expression in several conditions [78].	Each node is assigned an aggregated z-score calculated over several conditions.	If node degree is greater than a user-defined parameter d_{min}, all neighbours of the node that are not in the top-scoring component are removed.	–
Heuristic based on breadth and depth first search (Rajagopalan and Agarwal [139])	Undirected network integrating data from Ingenuity Pathways Knowledge Base, TRANSFAC and HumanCyc, thus covering metabolism, signal transduction and gene regulation (9,300 nodes and 30,000 edges).	None	Low weights for gene nodes of interest (randomly distributed between 0 and 10^{-3} to simulate low p-values). All other gene nodes received a weight randomly distributed between 0 and 1 (to simulate random p-values).	Exclusion (common cofactors, ATP, etc.) and penalty score for hub nodes. Penalty depends on a user-provided parameter β, which controls the size of the sub-network.	Undirected network: reaction directionality cannot be treated.
Heuristic and exact Steiner tree algorithms (Scott et al. [147])	Undirected protein-protein and protein-DNA interaction network integrating data from BIND, TRANSFAC, SCPD [176] and yeast ChIP-Chip experiments (5,458 nodes and 23,642 edges).	Gene expression in several conditions [78].	P-value p_u of differentially expressed gene u is assumed to be given. P_u is then converted into a weight by the weight function $w_{d(u)} = -log(1-p_u)$. An alternative weight function w_1 assigns weight of one to each node.	None	–
Heuristic based on merging filtered paths (Noirel et al. [119])	Undirected enzyme network obtained from KEGG (network size not indicated).	Ratio of protein quantities measured for two conditions [153]	Weight of an edge between two enzymes is the number of occurrences of connecting compound (for several compounds, least common one is used).	Weights (hub compounds receive a penalty)	Undirected network: reaction directionality cannot be treated.
Exact Steiner tree algorithm (Dittrich et al. [40])	Undirected protein-protein interaction network obtained from HPRD [136] (9,392 nodes and 36,504 edges for full network, intersection network with gene-expression study: 2,561 nodes and 8,538 edges)	Gene expression (B-cell lymphomas, [142]) and patient survival data (190 patients, [142])	P-values derived from different sources are aggregated using ith order statistic. The p-value distribution is assumed to be a mixture of signal and noise. The noise distribution is assumed to be uniform, whereas the signal distribution is modeled as a β distribution. The mixed distribution depends on one shape parameter (from the β distribution) and a mixture parameter. Both parameters are obtained from the p-values by numerical optimization. The score of a p-value is the log ratio between its value under the signal and the noise distribution. For a given false-discovery rate, a threshold p-value is computed. The score is then adjusted with this threshold p-value. The false-discovery rate is a user-provided parameter which controls the size of the sub-network.	None	–

Table 1.9: Multiple-end pathway prediction strategies - First part.

Subgraph extraction strategy (reference)	Biological network	Omics data mapped on network	Network weight assignment	Treatment of hub nodes	Metabolic networks only: treatment of reaction directionality
Heuristic based on step-wise integration of neighbours in sub-network (Antonov et al. [3])	Undirected metabolic network obtained from KEGG (network size not indicated).	None	None	KEGG RPAIR (only main reactant pairs are accepted)	Undirected network: reaction directionality cannot be treated.
Heuristic Steiner tree algorithms and random walks-based sub-network extraction (Faust et al. [55])	Directed and undireced metabolic networks obtained from MetaCyc (15,607 nodes, including duplicated reaction nodes, 43,938 edges), KEGG LIGAND (20,789 nodes, including duplicated reaction nodes, 61,726 edges) and KEGG RPAIR (16,826 nodes and 44,236 edges).	None	A weight of one is assigned to each reaction node, whereas a weight equal to its degree is assigned to each compound node.	Weights (hub compounds receive a penalty), optionally KEGG RPAIR	A reaction is represented by two nodes in the metabolic network: one for the forward and another one for the reverse direction of the reaction. Optionally, the reverse direction may be omitted to represent irreversible reactions.

Table 1.10: Multiple-end pathway prediction strategies - Second part.

Reference	Algorithm	Sub-network score calculation	Computational validation	Availability of tool	Remarks
Ideker et al. [77]	Space of high-scoring sub-networks is searched with a simulated annealing algorithm.	Node weights of sub-network are summed to calculate sub-network score. Distribution of sub-network scores for a fixed gene number is calculated using a Monte Carlo approach. The sub-network score is corrected using the mean and standard deviation of this background score distribution. For multiple conditions, the vector of condition-specific sub-network scores is sorted and rank-adjusted by computing score significances with a binomial order statistic. The maximum score is then corrected with the background score distribution and returned as the sub-network's final score.	None (method is directly applied to the study cases).	Cytoscape plugin (jActiveModules)	Algorithm is not designed to handle seed genes, but weights may be adapted to distinguish genes of interest from other genes.
Rajagopalan and Agarwal [139]	Breadth-first search around positive-scoring nodes to create sub-networks combined with limited depth-first search to merge sub-networks.	As in [77], but with another correction of the sub-network score to reduce the sub-network size.	On 100 artificial pathways and 219 pathways from BioCarta.	None	Algorithm is not designed to handle seed genes, but weights may be adapted to distinguish genes of interest from other genes.
Scott et al. [147]	Dreyfus-Wagner [45] for exact solution (small seed node sets) and Klein-Ravi [95] for approximate solution (large seed node sets) of Steiner tree problem	Sum of sub-network node weights.	None (method is directly applied to the study cases).	The authors state that a VisANT plugin [76] was created, but this plugin could not be found. However, source code was available upon request from Nadja Betzler [15].	Algorithm is designed to handle seed genes (called distinguished genes by the authors).
Noirel et al. [119]	Paths are obtained by depth-first search around all up-regulated enzyme nodes. Paths are filtered according to the number of up-regulated enzymes they contain. Filtered paths are then merged to form a sub-network.	Sum of sub-network edge weights.	None (method is directly applied to a study case)	Authors state that routines in Perl are available, but they could not be found on the indicated web page.	Algorithm is not designed to handle seed genes, but weights may be adapted to do so.
Dittrich et al. [40]	Algorithm solving Steiner tree problem to optimality [105]	Sum of sub-network nodes.	Recovery of artificial pathways.	Heinz (Python script), requires commercial software to be installed (CPLEX)	Sub-optimal solutions with user-defined distance to optimal solution can be returned. Algorithm is not designed to handle seed genes, but weights may be adapted to do so.

Table 1.10: Multiple-end pathway prediction strategies - Second part.

Reference	Algorithm	Sub-network score calculation	Computational validation	Availability of tool	Remarks
Antonov et al. [3]	Seeds are connected with their first-degree neighbours and the largest among the resulting network components is returned as distance 1 sub-network. This procedure is repeated for distance 2, 3, ... so that a series of distance-specific sub-networks is created.	The score of a sub-network is the number of seeds connected by it. A background distribution of scores is computed k times for a fixed seed number. The p-value is then calculated as: $p = (n+1)/k$, where n is the number of scores in the background distribution equal or greater than the sub-network score.	A huge number of study cases (272) is presented, but no real validation is performed (which would allow to measure the accuracy of the procedure).	web server (http://mips.helmholtz-muenchen.de/proj/keggspider/ and http://mips.helmholtz-muenchen.de/proj/cmp/)	Algorithm is designed to handle seed genes or compounds. Separate web servers for genes (KEGGSpider [3]) and compounds (TICL [4]) in metabolic networks.
Faust et al. [55]	Approximative Steiner tree algorithms: Takahashi-Matsuyama [154], Klein-Ravi, repetitive REA (combination of k shortest paths). Random walk-based algorithm: kWalks [48, 21]. In addition, combinations of these algorithms were tested.	Sum of sub-network arc weights.	on 71 *S. cerevisiae* pathways from MetaCyc	web server (http://rsat.ulb.ac.be/neat/)	Algorithm is designed to handle seeds. Web server allows to submit custom networks with custom weights, thus algorithms can be combined with more sophisticated weight policies. For the pre-loaded metabolic networks, genes, EC numbers, reactions or compounds are accepted as seeds.

2 Two-end metabolic pathway prediction

Presented article:
K. Faust, D. Croes and J. van Helden
Metabolic Pathfinding Using RPAIR Annotation
Journal of Molecular Biology, vol. 388, pp. 390-414, 2009.

2.1 Introduction

The first step towards the prediction of metabolic pathways from multiple seeds is the increase of two-end path finding accuracy. As discussed in section 1.10.3, appropriate treatment of hub compounds (i.e. compounds involved in a large number of reactions) is crucial to reach high prediction accuracies. The key idea of the article presented here is to use pre-defined substrate-product pairs from the RPAIR database [97] in order to avoid side compounds.

Reactant pairs (RPAIRs) are manually annotated substrate-product pairs that map each substrate of a reaction onto its structurally most similar product (see Figure 1.19). RPAIRs are classified into main, trans, cofac, ligase or leave RPAIRs according to their role in the reaction (see Table 1.8).

There are different ways in which reactant pairs can be integrated into path finding. We therefore assessed the impact of a number of different parameters on path finding accuracy, namely network type, network directionality, RPAIR role filtering, compound filtering and different weight policies. Table 2.1 describes each of these parameters.

2.2 Contribution

D. Croes conceived the idea of integrating RPAIRS to improve path finding. I performed the evaluation, analyzed the study cases and wrote the article. J. van Helden supervised the work and substantially revised the article.

2.3 Methods

Three networks were constructed: one from KEGG LIGAND (termed reaction network) and two from KEGG RPAIR (termed RPAIR and reaction-specific RPAIR network). The reference pathways were taken from aMAZE [101]. Of these, 25 are branched and seven contain cycles.

Table 2.1: Table of parameters that were assessed during the evaluation of path finding

Parameter	Description
Network type	Network consisting of all reactions and compounds in KEGG LIGAND versus network consisting of all reactant pairs and compounds in KEGG RPAIR
Network directionality	Network is directed (where each reaction is represented by two direction nodes) or undirected
RPAIR role filtering	RPAIRS of a specific role (such as cofac, ligase or leave) are removed from the RPAIR network
Compound filtering	Highly connected compounds (such as ATP or H_2O) are removed from the network
Weight policy	Four weight policies were tested: 1) unit (all nodes receive a weight of one) 2) degree (all compound nodes receive a weight equal to their degree) 3) RPAIR weight (all reactant pairs receive a role-specific weight) 4) RPAIR weight and degree (combination of weight policies 2 and 3)

Path finding cannot predict pathways having more than two terminal nodes (i.e. nodes without incoming or outgoing arcs), because it accepts only two seed nodes. It was therefore necessary to linearize the reference pathways, i.e. to extract their linear parts. Details on the reference pathways and their treatment are listed in section 9.2.

To evaluate the accuracy of two-end pathway prediction, the start and end reaction of a reference pathway are given to the path finding algorithm, which is based on REA [83]. All first-ranked paths (i.e. paths of equal weight) are merged to form the predicted pathway. The path finding accuracy is then calculated on the basis of non-seed node overlap between the predicted and the reference pathway (see section 9.2).

Path finding was carried out under various conditions for a set of 55 reference pathways from three organisms.

2.4 Results

With the best-performing parameter combination, an overall accuracy of 83% was reached for the 55 reference pathways (93% for 32 *E. coli* pathways, 66% for 11 *S. cerevsiae* pathways and 70% for *H. sapiens* pathways).

The assessment of parameter impact revealed that:

- In agreement with [31, 32], a weighted metabolic network performs better than an unweighted or filtered network (where hub compounds are removed).
- The RPAIR network outperforms the reaction network.
- Directed reaction networks perform better than undirected ones, because they prevent the path finding algorithm to go from substrate to substrate or from product to product. In contrast, for RPAIR networks, there is no difference between directed and undirected networks, since each reactant pair has only one substrate and one product.
- Filtering of RPAIR classes does not increase the path finding accuracy, because some reference pathways contain other than main RPAIRs.
- Weighting of RPAIRs only increases path finding accuracy in the absence of compound weights.

2.5 Conclusion

Our evaluation has shown that the combination of reactant pairs with a weight policy penalizing hub compounds yielded the highest path finding accuracy. Thus, metabolic pathway prediction accuracy can be increased by taking into account RPAIR annotations.

doi:10.1016/j.jmb.2009.03.006 *J. Mol. Biol.* (2009) **388**, 390–414

JMB

Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Metabolic Pathfinding Using RPAIR Annotation

Karoline Faust*, **Didier Croes and Jacques van Helden**

Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe), Université Libre de Bruxelles, Campus Plaine, CP 263, Bld du Triomphe, B-1050 Bruxelles, Belgium

Received 9 November 2008; received in revised form 25 February 2009; accepted 3 March 2009
Available online
10 March 2009

Edited by M. Sternberg

Metabolic databases contain information about thousands of small molecules and reactions, which can be represented as networks. In the context of metabolic reconstruction, pathways can be inferred by searching optimal paths in such networks. A recurrent problem is the presence of pool metabolites (e.g., water, energy carriers, and cofactors), which are connected to hundreds of reactions, thus establishing irrelevant shortcuts between nodes of the network. One solution to this problem relies on weighted networks to penalize highly connected compounds. A more refined solution takes the chemical structure of reactants into account in order to differentiate between side and main compounds of a reaction. Thanks to an intensive annotation effort at KEGG, decompositions of reactions into reactant pairs (RPAIR) categorized by their role (main, trans, cofac, ligase, and leave) are now available.

The goal of this article is to evaluate the impact of RPAIR data on pathfinding in metabolic networks. To this end, we measure the impact of different parameters concerning the construction of the metabolic network: mapping of reactions and reactant pairs onto a graph, use of selected categories of reactant pairs, weighting schemes for compounds and reactions, removal of highly connected metabolites, and reaction directionality. In total, we tested 104 combinations of parameters and identified their optimal values for pathfinding on the basis of 55 reference pathways from three organisms.

The best-performing metabolic network combines the biochemical knowledge encoded by KEGG RPAIR with a weighting scheme penalizing highly connected compounds. With this network, we could recover reference pathways from *Escherichia coli* with an average accuracy of 93% (32 pathways), from *Saccharomyces cerevisiae* with an average accuracy of 66% (11 pathways), and from humans with an average accuracy of 70% (12 pathways). Our pathfinding approach is available as part of the Network Analysis Tools.

Keywords: pathfinding; KEGG RPAIR; metabolic network; pathfinding evaluation; metabolic pathways

Introduction

In biochemical textbooks[1] and in databases such as KEGG[2,3] or BioCyc,[4,5] metabolic information is represented in the form of generic or organism-specific metabolic maps or pathways. Historically, the characterization of metabolic pathways relied on a very few model organisms (*Escherichia coli*, *Salmonella*, *Saccharomyces cerevisiae*, mouse, human, and so on). Enzyme grouping into pathways was defined on the basis of mutant phenotypes (e.g., methionine auxotrophy) and, with progress in biochemistry, detailed successions of reactions could be established. Such biochemical analyses revealed that different organisms can use alternative pathways to produce or utilize the same molecules. For example, in *E. coli*, L-lysine is produced in nine reactions from L-aspartate, whereas in *S. cerevisiae*, it is produced in eight reactions from 2-oxoglutarate. The documented examples of alternative pathways probably reflect a tiny part of the huge diversity of biochemical pathways.

The large size of the biochemical pathway space is also apparent from calculations performed by

*Corresponding author. E-mail address: kfaust@ulb.ac.be.

Abbreviations used: Sn, sensitivity; PPV, positive predictive value; Acc, geometric accuracy.

Kueffner *et al.*, who constructed a PETRI net from several metabolic databases and enumerated all paths between glucose and pyruvate.[6] Without further constraints, they obtained around 500,000 paths. Even after application of a number of constraints, as many as 170 pathways connecting glucose to pyruvate remained.

This demonstrates that detecting *de novo* pathways by exhaustive enumeration of paths will return many false positives, whereas relying on previous knowledge will overlook a large number of valid pathways. Automated metabolic pathfinding approaches can assist biochemists by proposing a reasonable number of hypothetic pathways ranked by potential relevance, which may then be filtered and modified on the basis of biochemical knowledge and validated experimentally.

Metabolic pathfinding can be applied to predict pathways between enzymes encoded by genes that are assumed to be functionally related (on the basis of coexpression, operon organization, phylogenetic profiles, gene fusion, and so on). This is especially useful in metabolic reconstruction, whose goal is to decipher organism-specific metabolism from genome data. Reconstruction strategies can be classified according to their extent of automation: In the absence of automation, data obtained from metabolic literature, expert knowledge, or databases are manually assembled into a metabolic network.[4,7,8] Once a set of pathways is known, the reconstruction procedure can be automated, using known pathways as template. The reconstruction procedure can be entirely automated[9–14] or can rely on a computer-based assignment refined by manual annotation as in "Tier 2" BioCyc databases.[12] Building a network from known pathways alone suffers from a serious limitation: The reconstructed network is restricted to previously characterized pathways. This does not allow the discovery of new alternatives. This limitation becomes apparent when reconstructed pathways contain gaps (i.e., reactions not catalyzed by any enzyme annotated in the query genome). A gap may occur because (1) the pathway is absent from the organism of interest and reactions identified for it belong to other pathways; (2) the enzyme-coding gene has not yet been identified in the query genome; (3) the organism uses a variant of the pathway that bypasses the gap (reasons 1–3 are listed in Paley and Karp[15]); or (4) the reaction is spontaneous in this organism. Metabolic pathfinding helps to address this problem: On one hand, it can suggest alternatives to the pathway in question and, on the other hand, it can be applied to identify and rank candidate gap filler enzymes.[16,17]

Metabolic pathfinding relies on a metabolic network (equivalently called "metabolic graph") where compounds and reactions are nodes linked by edges representing substrate/product relationships.[6,18,19] Various algorithms can be used to find one (shortest pathfinding) or several (*k*-shortest pathfinding) paths between a given pair of start and end nodes. Unfortunately, the shortest paths found in a raw metabolic network generally do not correspond to relevant biochemical pathways.[11,20,21] Indeed, a major problem of metabolic pathfinding is the presence of compounds involved in a high number of reactions. Typically, these compounds are cofactors (e.g., $NADP^+$/NADPH and NAD^+/NADH), small inorganic molecules (e.g., H_2O, O_2, and CO_2), or energy carriers (e.g., ATP and ADP). The shortest pathfinding algorithms will use such highly connected compounds as shortcuts and will thus infer irrelevant pathways containing cofactors or energy carriers as intermediates (Fig. 1a).

Failure to deal with highly connected compounds will bias any topological analysis of metabolic networks that is based on pathfinding. For instance, Jeong *et al.* described a "small-world" property of metabolic networks, which states that each compound in the network can be reached from other compounds in a small number of steps.[18] However, the "small-world" property was questioned by several authors because most of these short paths result from shortcuts linking two reactions via some irrelevant compounds.[11,20–22]

A number of strategies have been devised to overcome the problem of highly connected compounds.

The concept of "pool compounds" has been introduced. Compounds in the pool are freely available to a reaction in the pathway, whereas other compounds have to be produced or consumed by the pathway. This concept is especially applied in flux balance analysis where pool compounds are called external metabolites, which do not need to be balanced (e.g., Schuster *et al.*,[23] Teusink *et al.*,[24] and Edwards and Palsson[25]).

Karp and Paley defined "main compounds" as those shared between subsequent reaction steps of a pathway: They form the "backbone" of that pathway.[26] In contrast, "side compounds" are not involved in subsequent reaction steps.

Various authors[16,19,22] have excluded a selected list of compounds from the metabolic network under study. However, since a good definition of pool compounds is missing, it is unclear which compounds are to be removed. In addition, removal of those compounds generally avoids irrelevant shortcuts, but occasionally prevents the finding of some pathways in which they are used as intermediates (e.g., ADP is a side compound in most pathways, but a main compound for the *de novo* biosynthesis of purine nucleotides).

Another strategy is to take into account the chemical structure of the compounds in order to trace, for each reaction, the transfer of atom groups between substrates and products.[20,27,28] This assumes a reaction-specific definition of "main" and "side" substrates/products. Arita's concept defines "a pathway from *X* to *Y* [as] a sequence of biochemical reactions through which at least one carbon in *X* reaches *Y*." This strategy of atom mapping and tracing introduced by Arita was applied in modified forms for pathfinding.[29–32] It requires knowledge of the structure of each compound and a tedious annotation of atom transfers within each reaction.

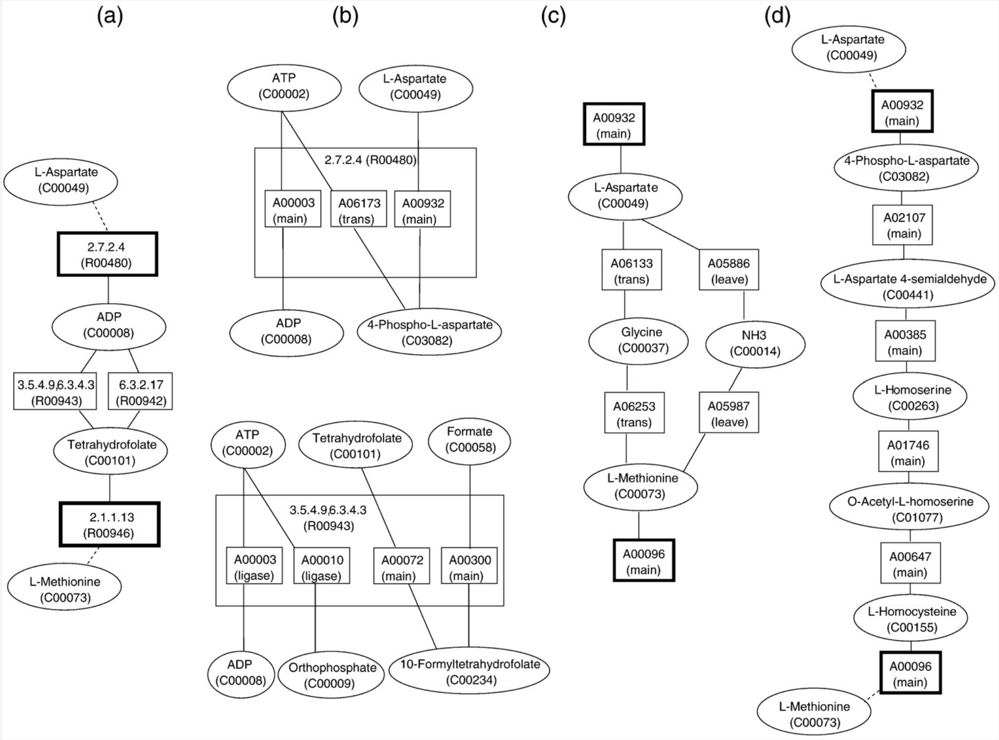


Fig. 1. Illustration of the relevance of RPAIR annotations. (a) Example showing that paths traversing pool metabolites often result in biochemically invalid pathways. The shortest path found in the unweighted reaction graph erroneously suggests that L-methionine can be produced from L-aspartate in three reaction steps via ADP and tetrahydrofolate (rectangles representing seed nodes have a border in boldface). (b) RPAIR composition of selected reactions used in the path of (a). Reaction R00480 is divided into three reactant pairs: The two reactant pairs labeled as "main" go from ATP to ADP, or from L-aspartate to 4-phospho-L-aspartate. In reaction R00943, the main reactant pairs connect tetrahydrofolate, formate, and 10-formyltetrahydrofolate. Note that there are no reactant pairs from L-aspartate to ADP or from ADP to tetrahydrofolate in these reactions, so that path (a) cannot be formed. (c) The shortest path between L-aspartate and L-methionine found in the unweighted RPAIR graph. (d) The shortest path between L-aspartate and L-methionine found in the weighted RPAIR graph.

Pool compounds can also be avoided by defining a set of rules,[33] assigned by experts on the basis of compound structure and biochemical literature, in order to enumerate all conversions allowed for each compound. The drawback of this approach is that the rule set might be incomplete or too restrictive to allow discovery of new pathways.

Kotera *et al.* introduced the concept of reactant pairs (RPAIRs), defined as "pairs of compounds that have atoms or atom groups in common on two sides of a reaction."[34,35] RPAIR thus establishes pairwise relationships between one substrate and one product, taking into account the respective roles of these compounds in the reaction. Accordingly, reactant pairs are grouped into the following classes: main, trans, cofac, ligase, and leave. For instance, a cofac reactant pair couples a substrate cofactor with a product cofactor (i.e., A00002 pairs NADH with NAD^+). Figure 1 shows typical examples of decomposition of reactions (Fig. 1a) into reactant pairs (Fig. 1b). This figure also demonstrates that biochemically irrelevant connections between ATP/ADP and other compounds are avoided in the RPAIR graph (Fig. 1c and d) simply because corresponding reactant pairs are absent in the RPAIR database. Note that the same reactant pair may take on different roles in different reactions (i.e., the ATP/orthophosphate reactant pair belonging to all five classes).

Oh *et al.* combined RPAIRs and structural comparisons to predict biodegradation pathways by iteratively matching a compound with its most similar partner in a library of reactant pairs.[36]

In a previous article, we introduced a strategy for predicting metabolic pathways based on *k*-shortest pathfinding in a weighted graph.[37] By default, each compound is assigned a weight equal to its degree ("connectivity") in the metabolic graph. The weight is then used as a penalty for pathfinding, leading to the concept of *lightest* pathfinding (as opposed to *shortest*). Thus, the more highly connected is a compound, the less likely it is to appear in an inferred pathway.

An important issue is the protocol used to assess the reliability of a pathway discovery method. Indeed, several approaches were tested on only a few study cases (e.g., McShan *et al.*,[29] Rahman *et al.*,[31] and de Figueiredo *et al.*[38]), or conclusions were drawn from global topological parameters without any attempt to compare discovered paths to annotated pathways (e.g., Jeong *et al.*[18]). In contrast, our strategy to penalize compounds according to their degree was supported by an evaluation on several tens of pathways.[21]

With the availability of the KEGG RPAIR database, we can now integrate manually compiled knowledge on atom flow as stored in the reactant pairs into pathfinding. In this article, we quantify the impact of KEGG RPAIR on pathfinding accuracy by carrying out a careful validation of known pathways from three organisms (*E. coli*, *S. cerevisiae*, and *Homo sapiens*). In addition, we evaluate the effects of parameters linked to the RPAIR database [e.g., the filtering and weighting of reactant pairs according to their class (main, trans, cofac, ligase, or leave)]. In order to compare results obtained with KEGG RPAIR to those obtained from our previous method, we also measure the impact of different weighting schemes and the filtering of pool metabolites on pathfinding accuracy.

We added the new version of the metabolic pathfinding tool and the best-performing metabolic networks to the Network Analysis Tools Web server.[39]

Results and Discussion

Graph construction and evaluation

We summarize here the main features of the metabolic graphs and pathways used for evaluation. The detailed description can be found in Materials and Methods.

We constructed three bipartite metabolic graphs from KEGG/LIGAND. The first one, called *reaction graph*, is built from all compounds and reactions, in the same way as in our previous work.[21] The second graph, named *RPAIR graph*, consists of all reactant pairs listed in KEGG RPAIR and their associated compounds. The third graph, called *reaction-specific RPAIR graph*, is a variant of the RPAIR graph, where each reactant pair instantiates one separate node for each reaction in which it is involved. Reactant pairs associated with the same reaction are mutually exclusive.

The performance of these graphs was evaluated on a set of 55 known reference pathways from three organisms (*E. coli*, *S. cerevisiae*, and *H. sapiens*). For each of the reference pathways, we measured how well it could be recovered from the metabolic graph given its start and end reactions alone.

This evaluation was repeated for each of the three graphs and for a variety of parameter values. The tested parameters included compound weighting scheme, reactant pair weighting scheme, graph directionality, compound filtering, and reactant pair filtering.

Study cases

In the following, we discuss three examples that illustrate the benefits of RPAIR annotations.

Two additional examples (lysine biosynthesis and proline degradation) demonstrate how pathfinding can uncover pathways in organism-specific networks and suggest alternatives to known metabolic pathways.

For each graph mentioned in these examples, we used parameter values yielding the highest pathfinding accuracies in our evaluation.

Aldosterone biosynthesis in humans

Aldosterone biosynthesis (Fig. 2a) in humans produces the hormone aldosterone from cholesterol. Aldosterone regulates the absorption of ions in the kidney.

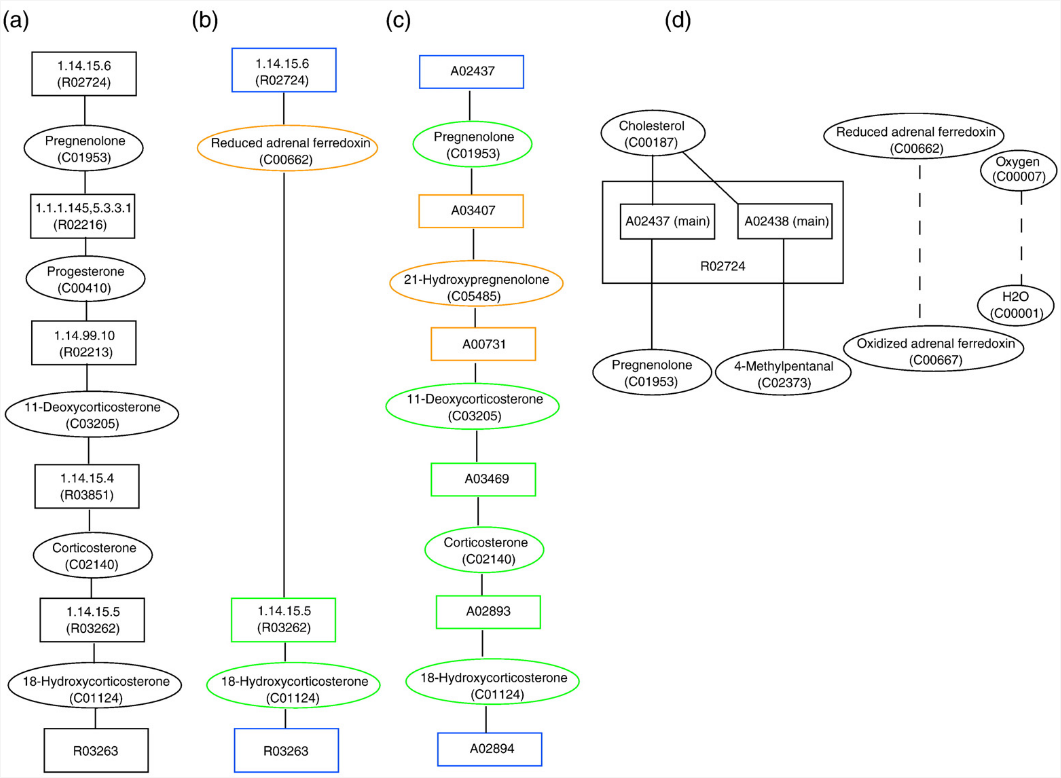


Fig. 2. Inference of the aldosterone biosynthesis pathway in the reaction graph and RPAIR graph. (a) Human aldosterone pathway as annotated in the aMAZE data set. (b) The lightest path found in the compound-weighted reaction graph. (c) The lightest path found in the compound-weighted RPAIR graph. (d) Reactant pair composition of the first reaction in the annotated pathway. Note the absence of any reactant pair connecting cholesterol to reduced adrenal ferredoxin. The compound pairs oxygen/water and reduced adrenal ferredoxin/oxidized adrenal ferredoxin differ by hydrogen atoms and electrons, respectively, which are not covered by KEGG RPAIR. Green, true positives; orange, false positives; blue, seed nodes.

With the compound-weighted reaction graph, we predict a short path (Fig. 2b) that skips a major part of the reference pathway by going through a side compound (reduced adrenal ferredoxin). The degree-based weighting scheme does not deal well with this compound because it appears in a very few reactions only (in four reactions as side compound).

In the RPAIR graph, we correctly recover the second half of the pathway, but the first half of the inferred path bypasses progesterone by 21-hydroxypregnenolone (Fig. 2c). This difference between reaction and RPAIR graph is mainly due to the absence of a reactant pair that couples reduced adrenal ferredoxin with cholesterol. Figure 2d depicts the reactant pair composition of the reaction concerned (R02724) showing that reduced adrenal ferredoxin and cholesterol do not exchange atoms according to RPAIR annotation.

Despite its deviation from the annotated pathway and in contrast to the path found in the reaction graph, the path obtained from the RPAIR graph makes sense biochemically. This alternative pathway to aldosterone is annotated in the corresponding KEGG map (C21—steroid hormone metabolism) and is also mentioned by Pasqualini as a way to generate deoxycorticosterone in human placenta.[40]

Furthermore, the reference pathway is found in the RPAIR graph as second-ranked path and in the reaction graph as fifth-ranked path.

Arginine biosynthesis in E. coli

The arginine biosynthesis pathway produces arginine from glutamate (Fig. 3a). In the compound-weighted reaction graph (Fig. 3b), a path that directly connects *N*-acetyl-L-glutamate to *N*-acetylornithine (reaction R02282) is found, bypassing *N*-acetyl-L-

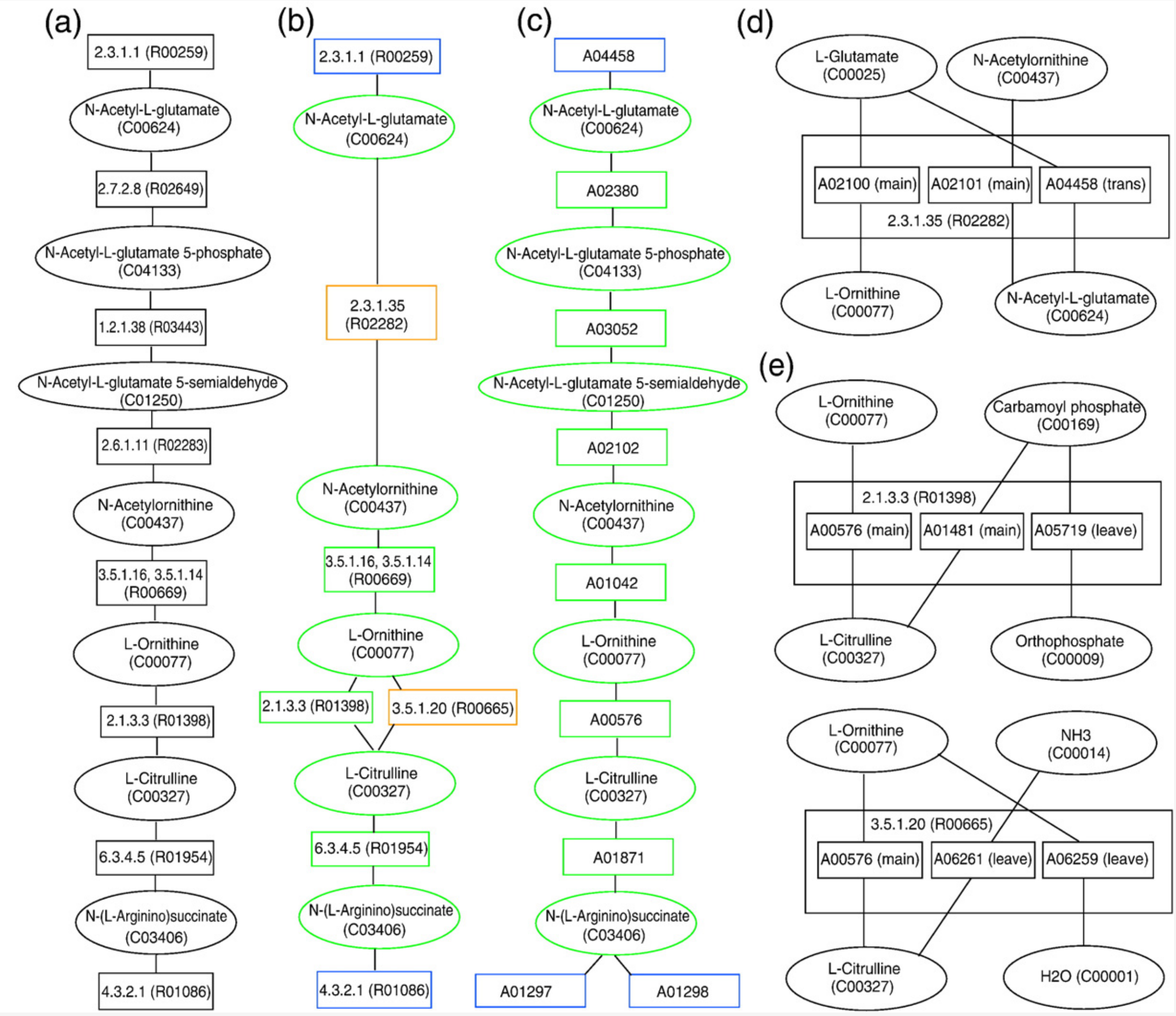


Fig. 3. Inference of the arginine biosynthesis pathway in the reaction graph and RPAIR graph. (a) Arginine biosynthesis pathway as annotated in the aMAZE data set for *E. coli*. (b) The lightest path found in the compound-weighted reaction graph. (c) The lightest path found in the compound-weighted RPAIR graph. (d) RPAIR composition of reaction R02282, used as a shortcut in (b). This reaction shares a reactant pair (A04458) with the start reaction, which is the reason for its avoidance in the RPAIR graph. (e) RPAIR composition of reactions R01398 and R00665, which give alternative connections in (b). Green, true positives; orange, false positives; blue, seed nodes.

glutamate 5-phosphate and *N*-acetyl-L-glutamate 5-semialdehyde.

This shortcut is avoided in the RPAIR graph (Fig. 3c) because the starting reactant pair A04458 excludes all reactant pairs (A02100, A0201, and A04458) associated with the bypass reaction (R02282) of Fig. 3b. In Fig. 3d, the reactant pair composition of the bypass reaction is shown.

In addition, the pathway inferred in the reaction graph consists of two alternative paths: one connecting L-ornithine to L-citrulline via reaction R01398 and another via reaction R00665. Both reactions differ by their side compounds: In the annotated reaction R01398, carbamoyl phosphate is added to L-ornithine by the enzyme carbamoyl transferase. The alternative reaction R00665 is catalyzed by the enzyme citrullinase, which hydrolyzes citrulline to form ornithine, ammoniac, and water. Citrullinase favors the formation of ornithine from citrulline more strongly than the reverse. The reactant pair composition of both reactions is depicted in Fig. 3e.

By default, alternative reactions in inferred pathways are counted as false positives. However, in the RPAIR graph, we can no longer differentiate between the two reactions. Both share a common main reactant pair, namely, A00576, which connects L-ornithine and L-citrulline. This loss of one false positive in the inferred path contributes to the higher accuracy reached with the RPAIR graph.

Pyruvate oxidation pathway in E. coli

The first reaction (R03145) of the pyruvate oxidation pathway in *E. coli* converts pyruvate into acetate (Fig. 4a) and can be divided into four reactant pairs (A00473, A02797, A05678, and

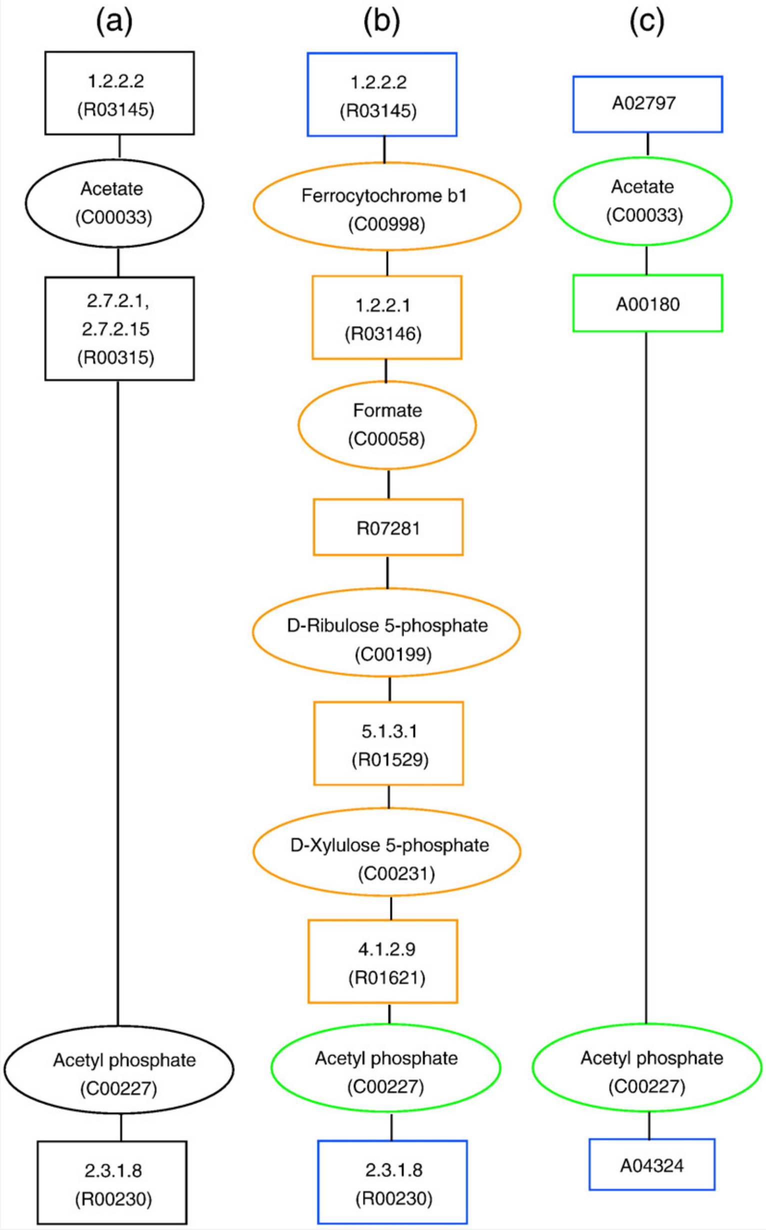


Fig. 4. Inference of the pyruvate oxidation pathway in the reaction graph and RPAIR graph. (a) Pyruvate oxidation pathway as annotated in the aMAZE data set for *E. coli*. (b) The lightest path found in the compound-weighted reaction graph. (c) The lightest path found in the compound-weighted RPAIR graph. Green, true positives; orange, false positives; blue, seed nodes.

A05698). In the reaction graph, the correct path is missed because instead of following the main reactant pair A02797, the cofac reactant pair connecting ferrocytochrome *b*1 to ferricytochrome *b*1 is chosen (Fig. 4b). In the RPAIR graph, this pathway is inferred correctly (Fig. 4c).

Lysine biosynthesis

Metabolic pathfinding can suggest alternative pathways either by finding multiple paths in a generic metabolic network (the *k*-lightest paths) or by considering the top-ranking paths in different organism-specific networks. In Croes *et al.*, we showed that the top-ranking paths connecting L-aspartate and L-lysine obtained from the generic compound-weighted reaction network correspond to alternative pathways that are valid in different organisms.[21] However, without further information, it is unclear which path might be active in which organism. Here, we repeated the search in organism-specific RPAIR networks constructed from KEGG PATHWAY 46.0 for the three yeast species *S. cerevisiae*, *Saccharomyces bayanus*, and *Schizosaccharomyces pombe*, and for the prokaryotes *E. coli*, *Salmonella typhimurium*, and *Bacillus subtilis*. For each of the organism-specific networks, we performed a search between L-aspartate and L-lysine, as well as a search between acetyl-CoA and L-lysine (L-aspartate and acetyl-CoA are both known start compounds for lysine biosynthesis). Figure 5 displays the paths of first rank obtained for each organism, grouping together organisms with identical paths. For *E. coli*, *Sal. typhimurium*, and *B.*

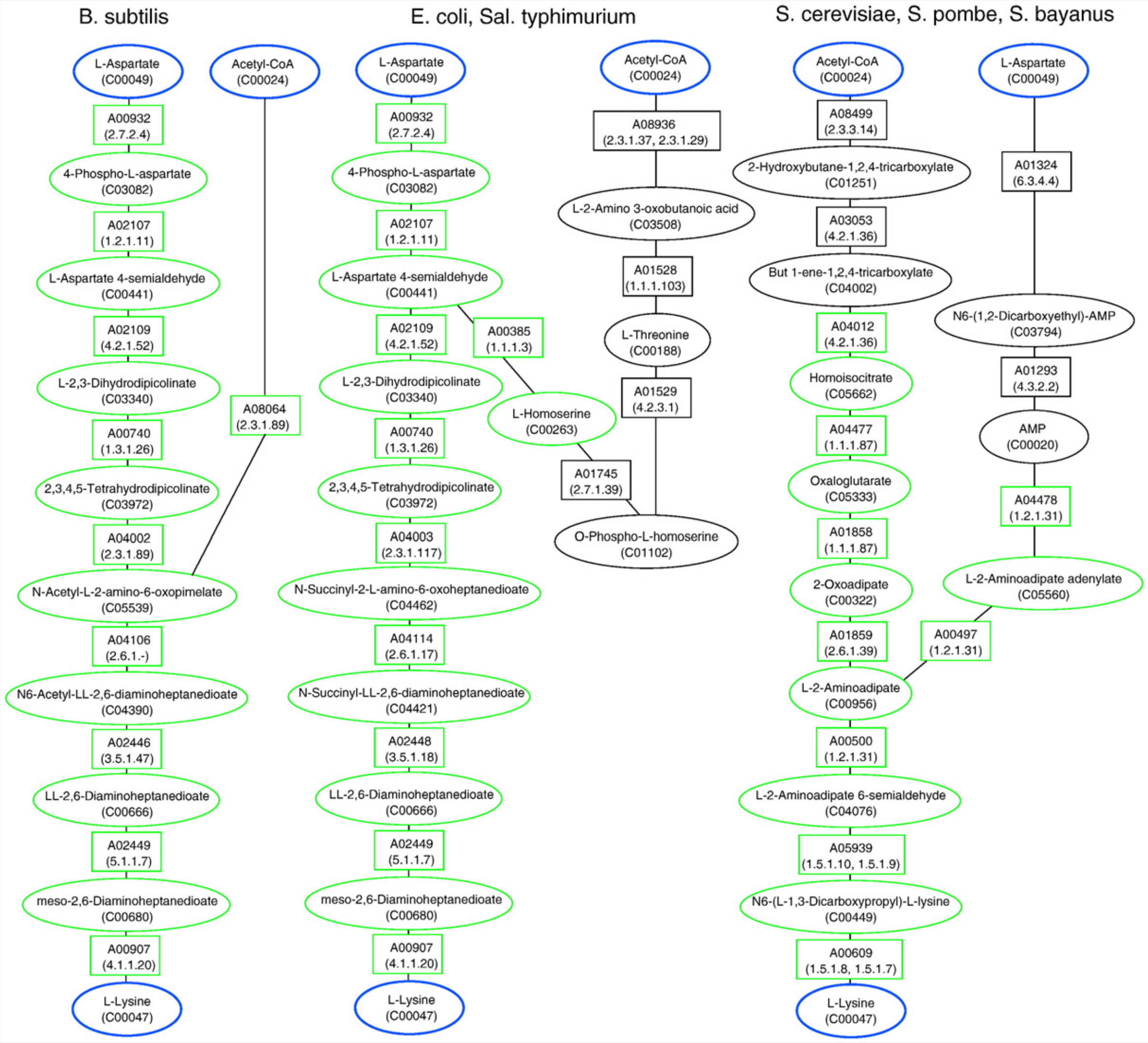

B. subtilis
E. coli, Sal. typhimurium
S. cerevisiae, S. pombe, S. bayanus
L-Aspartate (C00049)
Acetyl-CoA (C00024)
A00932 (2.7.2.4)
4-Phospho-L-aspartate (C03082)
A02107 (1.2.1.11)
L-Aspartate 4-semialdehyde (C00441)
A02109 (4.2.1.52)
L-2,3-Dihydrodipicolinate (C03340)
A08064 (2.3.1.89)
A00740 (1.3.1.26)
2,3,4,5-Tetrahydrodipicolinate (C03972)
A04002 (2.3.1.89)
N-Acetyl-L-2-amino-6-oxopimelate (C05539)
A04106 (2.6.1.-)
N6-Acetyl-LL-2,6-diaminoheptanedioate (C04390)
A02446 (3.5.1.47)
LL-2,6-Diaminoheptanedioate (C00666)
A02449 (5.1.1.7)
meso-2,6-Diaminoheptanedioate (C00680)
A00907 (4.1.1.20)
L-Lysine (C00047)
L-Aspartate (C00049)
A00932 (2.7.2.4)
4-Phospho-L-aspartate (C03082)
A02107 (1.2.1.11)
L-Aspartate 4-semialdehyde (C00441)
A02109 (4.2.1.52)
A00385 (1.1.1.3)
L-2,3-Dihydrodipicolinate (C03340)
L-Homoserine (C00263)
A00740 (1.3.1.26)
A01745 (2.7.1.39)
2,3,4,5-Tetrahydrodipicolinate (C03972)
A04003 (2.3.1.117)
N-Succinyl-2-L-amino-6-oxoheptanedioate (C04462)
A04114 (2.6.1.17)
N-Succinyl-LL-2,6-diaminoheptanedioate (C04421)
A02448 (3.5.1.18)
LL-2,6-Diaminoheptanedioate (C00666)
A02449 (5.1.1.7)
meso-2,6-Diaminoheptanedioate (C00680)
A00907 (4.1.1.20)
L-Lysine (C00047)
Acetyl-CoA (C00024)
A08936 (2.3.1.37, 2.3.1.29)
L-2-Amino 3-oxobutanoic acid (C03508)
A01528 (1.1.1.103)
L-Threonine (C00188)
A01529 (4.2.3.1)
O-Phospho-L-homoserine (C01102)
Acetyl-CoA (C00024)
A08499 (2.3.3.14)
2-Hydroxybutane-1,2,4-tricarboxylate (C01251)
A03053 (4.2.1.36)
But 1-ene-1,2,4-tricarboxylate (C04002)
A04012 (4.2.1.36)
Homoisocitrate (C05662)
A04477 (1.1.1.87)
Oxaloglutarate (C05333)
A01858 (1.1.1.87)
2-Oxoadipate (C00322)
A01859 (2.6.1.39)
L-2-Aminoadipate (C00956)
A00500 (1.2.1.31)
L-2-Aminoadipate 6-semialdehyde (C04076)
A05939 (1.5.1.10, 1.5.1.9)
N6-(L-1,3-Dicarboxypropyl)-L-lysine (C00449)
A00609 (1.5.1.8, 1.5.1.7)
L-Lysine (C00047)
L-Aspartate (C00049)
A01324 (6.3.4.4)
N6-(1,2-Dicarboxyethyl)-AMP (C03794)
A01293 (4.3.2.2)
AMP (C00020)
A04478 (1.2.1.31)
L-2-Aminoadipate adenylate (C05560)
A00497 (1.2.1.31)

Fig. 5. Pathfinding results for lysine biosynthesis in organism-specific compound-weighted RPAIR networks. Organisms for which identical paths were found are grouped together. The search was conducted between L-aspartate and L-lysine and between acetyl-CoA and L-lysine; the first-ranking paths obtained were then merged. For *B. subtilis* and *E. coli/Sal. typhimurium*, the complete annotated lysine biosynthesis pathways are covered; for the yeast species, a large part of the known pathway is found. Green, RPAIR/compound is a part of the lysine biosynthesis KEGG map; blue, seed nodes.

subtilis, the lysine biosynthesis pathways starting from L-aspartate are fully covered. For each yeast species, a large part of the known pathway starting from acetyl-CoA is found. Note that the Gram-negative bacteria *E. coli* and *Sal. typhimurium*, the Gram-positive bacterium *B. subtilis*, and the eukaryotes *S. cerevisiae*, *S. bayanus*, and *Sc. pombe* each employ a different lysine biosynthesis pathway.

Proline degradation

The following study case illustrates how alternatives to known pathways can be found by enumeration of paths in a generic metabolic network. The amino acid L-proline is degraded to L-glutamate. MetaCyc[5] lists two known pathways for proline degradation. The arginine and proline metabolism KEGG map also suggests two proline degradation pathways: one connecting L-1-pyrroline-5-carboxylate directly to L-glutamate and the other going through L-glutamate 5-semialdehyde before reaching L-glutamate. If we enumerate paths between L-proline and L-glutamate in the compound-weighted KEGG RPAIR network, we obtain these pathways as the first-ranked and second-ranked paths (Fig. 6, paths 1 and 2.1). In addition, several paths contained neither in MetaCyc nor in KEGG are returned (Fig. 6, other paths), suggesting the degradation of proline via D-proline and aminopentanoate (paths 2.2 and 3.1), via *trans*-4-hydroxy-L-proline (path 3.2), and via L-ornithine (paths 5.1 and 5.2).

Impact of parameter values

So far, we discussed the behavior of pathfinding in a few illustrative study cases. In order to assess the generality of our observations, we extend this analysis to a benchmark collection of 55 pathways from three model organisms: the bacterium *E. coli* (32 pathways), the yeast *S. cerevisiae* (11 pathways), and humans (12 pathways). We systematically tested the impact of six different parameters on

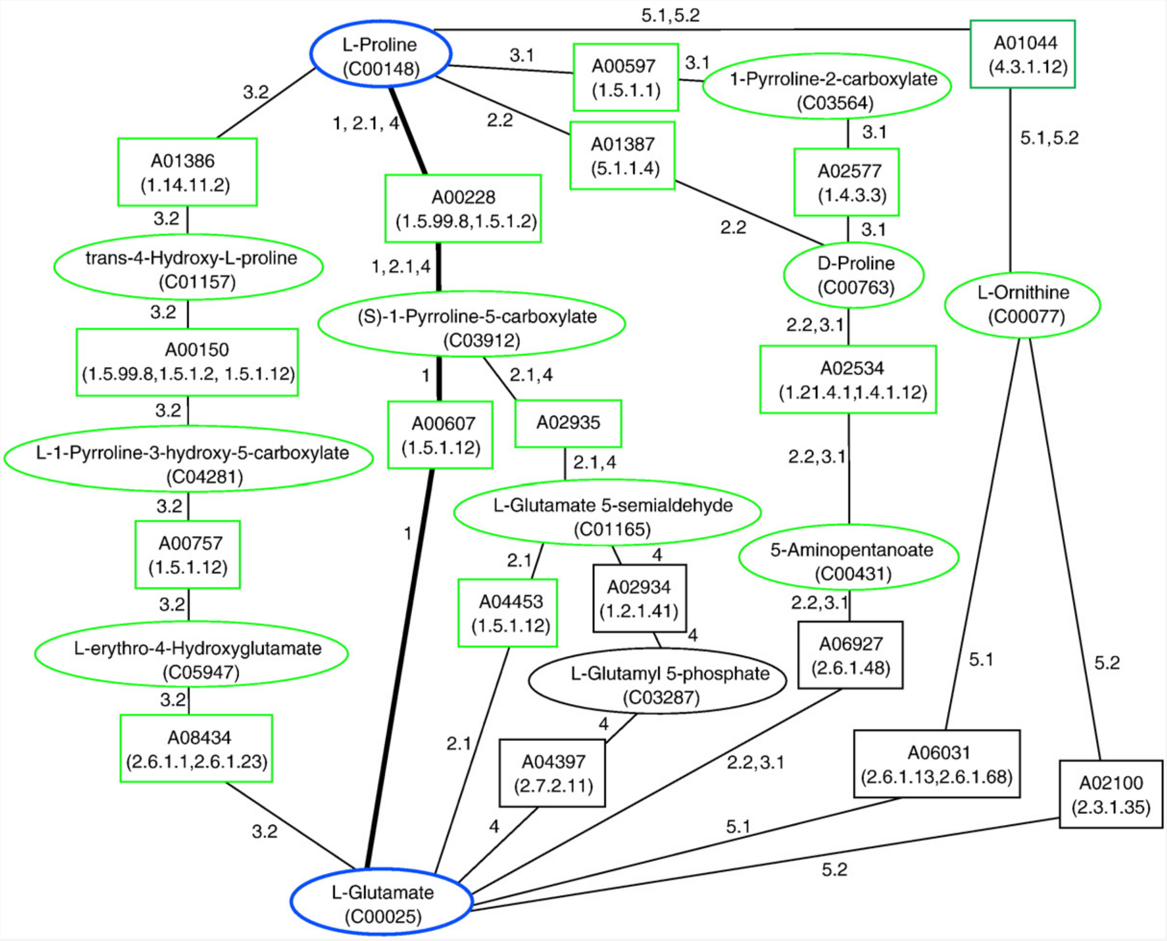


Fig. 6. Enumeration of proline degradation pathways in the compound-weighted RPAIR network. The top-ranking paths correspond to the annotated pathways for proline degradation, as listed in MetaCyc (path 1 highlighted with thick edges) and as visible from the arginine and proline metabolism KEGG map (paths 1 and 2.1). The paths of higher rank propose alternative pathways for the degradation of L-proline that are not apparent from the corresponding KEGG map. Green, RPAIR/compound is part of the arginine and proline metabolism KEGG map; blue, seed nodes. Edge labels: Rank index of the path(s) to which edge belongs. Paths of equal weights have the same rank index (first digit). The second digit distinguishes between paths of equal rank (e.g., paths 2.1 and 2.2 both have a weight of 130).

Table 1. Description of the parameters needed to construct the metabolic graphs

Parameter name	Values	Description
Graph type	Reaction	Bipartite graph with one node per compound and one node per reaction
	RPAIR	Bipartite graph with one node per compound and one node per reactant pair
	Reaction-specific RPAIR	Same as RPAIR, but a separate node is created for each combination of RPAIR/reaction
Graph directionality	Directed	Reaction and RPAIR nodes are duplicated in order to represent he two possible directions (forward and reverse); edges are directed
	Undirected	One node per reaction; undirected edges
Compound filtering	Filtered	36 highly connected compounds removed from the graph[21,26]
	Unfiltered	All compounds present in the graph
Compound weight	Unit	Each compound has the same weight ($w=1$)
	Degree	Each compound node has a weight equal to its degree (incoming + outgoing edges)
RPAIR class filtering	All	All LIGAND reactant pairs are present in the graph
	Main	The graph only contains reactant pairs classified as "main" + their substrates/products
	Main + trans	The graph only contains reactant pairs classified as "main" or "trans" + their substrates/products
RPAIR weight	Unit weight	Each RPAIR node has the same weight ($w=1$)
	Class-specific weight (only applies to RPAIR and reaction-specific RPAIR graphs)	Weights are assigned to reactant pairs according to their class as follows: main = 1; trans = 5; cofac = 10; ligase = 15; and leave = 20; if an RPAIR is found with several classifications in different reactions, we always choose the highest weight (the least favorable classification)

pathfinding accuracy (see Table 1 for a description of the parameters). Taking into account some interdependencies between parameters (e.g., reaction weights only apply to RPAIRs and reaction-specific RPAIRs), the total number of possible parameter combinations is 104. For each combination of parameters, we ran the *k*-shortest pathfinding algorithm on each test pathway and measured sensitivity (Sn), positive predictive value (PPV), and geometric accuracy (Acc) as described in Materials and Methods. The analysis of all pathways with one specific combination of parameters is referred to as "an experiment." We thus performed 104 distinct experiments.

Optimal combinations of parameters

Table 2 lists, for some selected experiments, the Acc averaged over all reference pathways in *E. coli*, *S. cerevisiae*, and *H. sapiens*, respectively (the complete results of the 104 experiments are available in Tables S1–S4). For all three species tested, the RPAIR graph, in combination with compound degree weights, yielded the highest Acc. The average Acc values reached for the top-ranking parameter combination were 93% for *E. coli*, 66% for *S. cerevisiae*, and 70% for human pathways, and 83% for all the pathways merged. There are three explanations for these organism-specific differences in accuracy. First, there is a selection bias due to the available annotations. Secondly, the pathway sets have different length distributions. For instance, the 12 human pathways analyzed here have an average length of 14 nodes, whereas the 32 *E. coli* pathways consist, on average, of only eight nodes. Thirdly, pathways are easier to infer if they are less interconnected with the metabolic network. In our selection, the human pathways happen to be more specialized (e.g., hormone metabolism) than the ones from *E. coli* and *S. cerevisiae*. This might be the reason that we reach a higher average accuracy for the human pathways compared to the yeast pathways, although the human pathways are, on average, longer than the yeast pathways. Despite these organism-specific differences, the following general trends appear in the ranking of experiments: Firstly, the 16 top-ranking combinations of parameters (Table S4) are always based on pathfinding in the RPAIR graph, with compound weighting. Among those, unit RPAIR weights always give better results than class-specific RPAIR weights. RPAIR class filtering does not appear to improve the results (the difference between "all RPAIRs" and "main + trans RPAIRs" is not significant; Table 3).

Statistical significance of parameter impact

In order to quantify the impact of each parameter, we performed a systematic comparison between each pair of alternative parameter values (e.g., degree *versus* unit compound weight), with all other parameters being equivalent ("experimentwise" comparison; Table 3). For a given pair of parameter values, we counted the number of experiments where the accuracy is better, worse, or identical, respectively. The significance of the differences was estimated by computing the *P*-value with a paired Wilcoxon signed-rank test. This systematic comparison shows that the most significant parameters are compound weight (degree weight is better than unit weight), graph type (RPAIR is better than reaction-specific RPAIR, and both are better than reaction graph), and compound filtering (filtered is better than nonfiltered).

It should be noted that, for each pair of parameter values, this comparison includes all the possible combinations of the five other parameters. However, because of interdependencies between some parameters, a given choice might return better or worse results, depending on the values chosen for the other parameters. In order to estimate the best possible performances for each parametric choice, we performed a second comparison, where each parameter value is coupled to its "best-friend" parameters (i.e., the combination of values for the five other parameters that returns the highest average accuracy for the reference pathways) ("best-combination-wise" comparison; Table 4). Interestingly, this analysis indicates a significant impact for one parameter only, the graph type: RPAIR outperforms both reaction graph ($P=2.4\text{E}-5$ for all pathways merged) and reaction-specific RPAIR graph ($P=4.7\text{E}-5$). At first sight, it might seem surprising that, in this best-combination-wise analysis (Table 4), the compound weighting (degree or unit) had no significant effect ($P=0.15$), whereas it appeared as the most significant parameter in the systematic comparison discussed above (Table 3). This is explained by the fact that the disadvantage of the unit weight can be compensated for by a specific combination of companion parameters, including the filtering of 36 highly connected compounds and the class-specific weighting of reactant pairs. It should also be noted that, for almost all pairs of parameters, the common best-friend parameter values include compound degree weight, thereby confirming the importance of this parameter for obtaining optimal accuracy.

In the following sections, we will try to understand the reason why these parameters affect the relevance of the inferred pathways.

Graph type

RPAIR graph versus reaction graph. We noticed the following reasons for accuracy differences between the RPAIR graph and the reaction graph:

1. Using the RPAIR graph eliminates false positives introduced by alternative reactions linking the same pair of compounds (e.g., R01398 and R00665; Fig. 3b). They are represented as a single reactant pair in the RPAIR graph (A00576; Fig. 3c).
2. A main compound and a side compound can be connected by a reaction in the reaction graph, without forming a reactant pair in the RPAIR graph, because the side compound only exchanges electrons, protons, or energy with the considered main compound (e.g., reduced adrenal ferredoxin and pregnenolone in reaction R02724 of aldosterone biosynthesis in humans; Fig. 2d).
3. A wrong path can be avoided in the RPAIR graph, thanks to the mutual exclusion of RPAIRs (e.g., reactant pair A04458 is associated with start reaction R00259 and reaction R02282, which prevents the traversal of reactant pairs belonging to R02282 in the RPAIR graph, as shown in Fig. 3 for arginine biosynthesis in *E. coli*).
4. In the RPAIR graph, start and end nodes are selected by identifying the main reactant pair of the initial and terminal reactions (e.g., RPAIR A02797 in the start reaction R03145 of pyruvate oxidation in *E. coli*; Fig. 4), thereby restricting the initial and final possibilities for searching inappropriate paths.
5. The RPAIR and reaction graphs differ in compound node degrees and, consequently, in compound weights.

Point 1 can be considered as a side effect of the scoring scheme. When calculating the accuracy of pathfinding with a scoring scheme that does not count as false positive the alternative reactions linking the same pair of compounds, the accuracy difference between the reaction graph and the RPAIR graph is reduced, but remains significant (Tables S5 and S6). Thus, the superior performance of the RPAIR graph cannot be explained solely by a side effect of the scoring scheme, but reflects the benefits of RPAIR annotation discussed in points 2, 3, and 4, and illustrated by the study cases. Concerning point 5, we found examples for pathways correctly recovered in the RPAIR graph, but not in the reaction graph, because of compound weight differences (i.e., arginine utilization in *E. coli*). Other paths are recovered in the reaction graph, but not in the RPAIR graph, for the same reason (i.e., aromatic amino acid biosynthesis in *S. cerevisiae*). We may assume that differences in compound weights favor neither the reaction graph nor the RPAIR graph.

RPAIR graph versus reaction-specific RPAIR graph. The mutual exclusion of reactant pairs is less strict in the reaction-specific RPAIR graph than in the RPAIR graph. To clarify this, consider the following example: The cofac RPAIR A00002, converting NAD^+ into $NADH+H^+$, occurs in 810 reactions. In the RPAIR graph, as soon as any of these reactions has been traversed through the main reaction, the associated cofac RPAIR is marked as forbidden in the next pathfinding steps. In contrast, in the reaction-specific RPAIR graph, the RPAIR A00002 is represented as 810 independent nodes. Let us assume that we traverse the main RPAIR of some reaction involving A00002 as cofac RPAIR. In the reaction-specific RPAIR graph, other instances of A00002 can still occur among the next path steps as part of another reaction. In the RPAIR graph, the cofac reactant pair A00002 is excluded from the path.

Under the default scoring scheme, the RPAIR graph performs significantly better than the reaction-specific RPAIR graph. However, this superior performance of the RPAIR graph comes at a cost: When inferring a path, we do not know which of several reactions sharing the same reactant pair has been predicted. The reaction-specific RPAIR graph gives a more precise answer, returning for each reactant pair its associated reaction. However, this apparent precision might be misleading because pathfinding may arbitrarily choose one among the possible reactions associated with a given RPAIR.

The difference in performance between RPAIR graph and reaction-specific RPAIR graph is less significant when applying the alternative scoring scheme (in which alternative reactions between compound pairs are not counted as false positives; see Tables S5 and S6).

Directionality

We do not observe a significant difference between directed and undirected graphs in any of the Wilcoxon tests.

When considering the reaction-specific RPAIR graph or the RPAIR graph, we realize that each reactant pair has, by definition, only one substrate and one product. This prevents traversion of a reaction from substrate to substrate or from product to product in these graphs. In contrast, for the weighted reaction graph, the directed graph performs better than the undirected one (Table 2).

Compound filtering and weighting

Consistent with our previous study,[21,37] the present evaluation shows that a compound-weighted graph yields higher accuracies compared to a compound-filtered graph and that the latter increases accuracy compared to the raw graph (Table 2). In the experimentwise Wilcoxon test, the impact of both filtering and weighting is highly significant (Table 3). As discussed above, this significance does not appear in the best-combination-wise Wilcoxon test (Table 4) because the “best-friend” parameters of the unfiltered or unweighted graph compensate for the absence of compound filtering or weighting. However, in this table, the compound degree weighting scheme is always among the best friends of all other parameter values.

Reaction filtering and weighting

The existence of annotated pathways containing trans reactant pairs shows that, analogous to the removal of highly connected compounds, the exclusion of non-main reactant pairs might cause the loss of valid pathways. Indeed, RPAIR graphs consisting of main reactant pairs perform worse than those including reactant pairs of classes main and trans or of all classes. The performance decreases not only because pathways with trans reactant pairs are missed but also because the weight of highly connected compounds is reduced by this strict filtering policy, thereby decreasing the efficiency of the compound degree weighting scheme.

To avoid the drastic effect of filtering, we tested a weighting scheme that maintains all RPAIR classes while favoring main reactant pairs over other reactant pairs (RPAIR class-specific weighting scheme). The difference in accuracy caused by this weighting scheme is not significant for any of the tested organisms in the best-combination-wise Wilcoxon test. In the experimentwise Wilcoxon test, a significant impact of reactant pair class-specific weights is only observed for *E. coli*.

Overall, neither filtering nor weighting of reactant pairs contributes significantly to pathfinding accuracy. It is possible that reactant pair class-specific weights can still be optimized, but such an optimization would require intensive testing.

Conclusion

In previous studies,[21,37] we evaluated the inference of metabolic pathways by pathfinding in three alternative graph types (raw, compound-filtered, and compound-weighted) and showed that pathfinding accuracy is strongly improved by weighting compounds according to their degree in the metabolic graph. We now extended this analysis by comparing 104 combinations of parameters and by quantifying the impact of each parameter on the accuracy of the inferred pathways. In particular, we assessed the benefit of a new level of annotation available in KEGG in the form of reactant pairs (RPAIRs). The main conclusion of our evaluation is that RPAIR annotation, when combined with compound weighting, significantly improves the quality of pathfinding. Our findings are consistent with a recent study that combines weighted metabolic graphs with atom mappings determined computationally.[32] In contrast, the present analysis relies on manual annotations by the KEGG/LIGAND team. From both studies, we may conclude that the highest accuracies are achieved when combining knowledge on atom flow in reactions (as provided by RPAIR) and a weighting scheme penalizing highly connected compounds.

Limitations of pathfinding

The *k*-shortest pathfinding relies on the assumptions that (1) the number of enzymes required to synthesize or degrade a compound has been minimized during evolution and that (2) each compound and each reaction appear only once in

Table 2. Impact of input parameters on the accuracy of pathway inference

Graph type	RPAIR class filtering	RPAIR weights	Compound weights	Directed	Filtered	Sn (%)	PPV (%)	Acc (%) Geometric accuracy
Organisms merged								
RPAIR graph	Main–trans	Unit	Degree	True	False	81.3	85.0	82.6
RPAIR graph	Main–trans	Unit	Degree	False	False	81.3	85.0	82.6
Reaction-specific RPAIR graph	All	Unit	Degree	True	False	79.6	75.3	76.7
Reaction-specific RPAIR graph	All	Unit	Degree	False	False	79.6	75.3	76.7
Reaction-specific RPAIR graph	All	Unit	Degree	True	True	79.2	75.6	76.7
Reaction-specific RPAIR graph	All	Unit	Degree	False	True	79.2	75.6	76.7
Reaction graph	**All**	**Unit**	**Degree**	**True**	**False**	**74.6**	**72.2**	**72.5**
Reaction graph	All	Unit	Degree	True	True	74.6	72.2	72.5
Reaction graph	All	Unit	Degree	False	False	71.2	69.9	69.5
Reaction graph	All	Unit	Degree	False	True	71.2	69.9	69.5
Reaction graph	**All**	**Unit**	**Unit**	**True**	**True**	**61.4**	**55.6**	**56.9**
Reaction graph	All	Unit	Unit	False	True	59.9	51.8	54.0
Reaction graph	**All**	**Unit**	**Unit**	**True**	**False**	**43.5**	**8.6**	**15.7**
Reaction graph	All	Unit	Unit	False	False	45.1	8.0	15.3
E. coli								
RPAIR graph	Main–trans	Unit	Degree	True	False	92.3	93.7	93.0
RPAIR graph	All	RPAIR class-specific	Degree	True	False	92.3	93.7	93.0
RPAIR graph	All	RPAIR class-specific	Degree	False	False	92.3	93.7	93.0
RPAIR graph	Main–trans	RPAIR class-specific	Degree	True	False	92.3	93.7	93.0
RPAIR graph	Main–trans	Unit	Degree	False	False	92.3	93.7	93.0
RPAIR graph	Main–trans	RPAIR class-specific	Degree	False	False	92.3	93.7	93.0
Reaction-specific RPAIR graph	Main–trans	RPAIR class-specific	Degree	False	True	92.0	83.0	87.0
Reaction graph	**All**	**Unit**	**Degree**	**True**	**False**	**85.7**	**78.0**	**81.3**
Reaction graph	All	Unit	Degree	True	True	85.7	78.0	81.3
Reaction graph	All	Unit	Degree	False	False	82.6	76.8	78.8
Reaction graph	All	Unit	Degree	False	True	82.6	76.8	78.8
Reaction graph	**All**	**Unit**	**Unit**	**True**	**True**	**73.9**	**66.1**	**69.2**
Reaction graph	All	Unit	Unit	False	True	73.0	63.1	66.5
Reaction graph	**All**	**Unit**	**Unit**	**True**	**False**	**51.8**	**9.5**	**18.6**
Reaction graph	All	Unit	Unit	False	False	53.1	8.3	17.7
S. cerevisiae								
RPAIR graph	All	Unit	Degree	False	True	59.5	76.4	65.5
RPAIR graph	All	Unit	Degree	False	False	59.5	76.4	65.5
RPAIR graph	All	Unit	Degree	True	False	59.5	76.4	65.5
RPAIR graph	All	Unit	Degree	True	True	59.5	76.4	65.5
Reaction-specific RPAIR graph	All	RPAIR class-specific	Degree	True	True	59.5	69.5	62.4
Reaction-specific RPAIR graph	All	RPAIR class-specific	Degree	True	False	59.5	69.5	62.4
Reaction-specific RPAIR graph	All	RPAIR class-specific	Degree	False	False	59.5	69.5	62.4
Reaction-specific RPAIR graph	All	Unit	Degree	False	True	59.5	69.5	62.4
Reaction-specific RPAIR graph	All	Unit	Degree	False	False	59.5	69.5	62.4
Reaction-specific RPAIR graph	All	RPAIR class-specific	Degree	False	True	59.5	69.5	62.4
Reaction-specific RPAIR graph	All	Unit	Degree	True	False	59.5	69.5	62.4
Reaction-specific RPAIR graph	All	Unit	Degree	True	True	59.5	69.5	62.4
Reaction graph	All	Unit	Degree	True	True	55.6	67.5	59.5
Reaction graph	**All**	**Unit**	**Degree**	**True**	**False**	**55.6**	**67.5**	**59.5**
Reaction graph	All	Unit	Degree	False	True	54.6	66.4	58.4
Reaction graph	All	Unit	Degree	False	False	54.6	66.4	58.4
Reaction graph	**All**	**Unit**	**Unit**	**True**	**True**	**45.5**	**52.5**	**46.7**
Reaction graph	All	Unit	Unit	False	True	40.3	46.8	42.6
Reaction graph	All	Unit	Unit	False	False	29.7	12.6	15.1
Reaction graph	**All**	**Unit**	**Unit**	**True**	**False**	**29.0**	**12.1**	**14.6**
H. sapiens								
RPAIR graph	Main–trans	Unit	Degree	True	True	70.0	71.4	70.2
RPAIR graph	All	Unit	Degree	False	True	70.0	71.4	70.2
RPAIR graph	All	RPAIR class-specific	Degree	False	False	70.0	71.4	70.2
RPAIR graph	Main–trans	RPAIR class-specific	Degree	False	True	70.0	71.4	70.2
RPAIR graph	All	RPAIR class-specific	Degree	True	True	70.0	71.4	70.2
RPAIR graph	All	Unit	Degree	False	False	70.0	71.4	70.2
RPAIR graph	Main–trans	Unit	Degree	False	False	70.0	71.4	70.2
RPAIR graph	Main–trans	RPAIR class-specific	Degree	False	False	70.0	71.4	70.2
RPAIR graph	All	Unit	Degree	True	False	70.0	71.4	70.2
RPAIR graph	Main–trans	Unit	Degree	True	False	70.0	71.4	70.2
RPAIR graph	All	RPAIR class-specific	Degree	True	False	70.0	71.4	70.2
RPAIR graph	Main–trans	Unit	Degree	False	True	70.0	71.4	70.2
RPAIR graph	All	RPAIR class-specific	Degree	False	True	70.0	71.4	70.2
RPAIR graph	Main–trans	RPAIR class-specific	Degree	True	True	70.0	71.4	70.2

Table 2 (*continued*)

Graph type	RPAIR class filtering	RPAIR weights	Compound weights	Directed	Filtered	Sn (%)	PPV (%)	Acc (%) Geometric accuracy
H. sapiens								
RPAIR graph	All	Unit	Degree	True	True	70.0	71.4	70.2
RPAIR graph	Main–trans	RPAIR class-specific	Degree	True	False	70.0	71.4	70.2
Reaction-specific RPAIR graph	All	RPAIR class-specific	Degree	False	True	68.0	60.6	64.1
Reaction-specific RPAIR graph	All	RPAIR class-specific	Degree	False	False	68.0	60.6	64.1
Reaction-specific RPAIR graph	All	RPAIR class-specific	Degree	True	False	68.0	60.6	64.1
Reaction-specific RPAIR graph	All	RPAIR class-specific	Degree	True	True	68.0	60.6	64.1
Reaction-specific RPAIR graph	All	Unit	Degree	True	False	68.0	60.6	64.1
Reaction-specific RPAIR graph	All	Unit	Degree	False	True	68.0	60.6	64.1
Reaction-specific RPAIR graph	All	Unit	Degree	True	True	68.0	60.6	64.1
Reaction-specific RPAIR graph	All	Unit	Degree	False	False	68.0	60.6	64.1
Reaction graph	**All**	**Unit**	**Degree**	**True**	**True**	**60.2**	**58.8**	**58.7**
Reaction graph	All	Unit	Degree	True	False	60.2	58.8	58.7
Reaction graph	All	Unit	Degree	False	False	52.9	51.9	51.9
Reaction graph	All	Unit	Degree	False	True	52.9	51.9	51.9
Reaction graph	**All**	**Unit**	**Unit**	**True**	**True**	**42.4**	**30.4**	**33.4**
Reaction graph	All	Unit	Unit	False	True	42.9	26.2	30.9
Reaction graph	All	Unit	Unit	False	False	38.0	3.0	9.2
Reaction graph	**All**	**Unit**	**Unit**	**True**	**False**	**34.6**	**3.0**	**8.7**

For each organism, experiments are sorted by decreasing Acc value. Due to space restrictions, we only display a subset of the conditions selected to highlight the impact of the most influential parameters. The complete tables of 104 experiments are available as Supplementary Material. We highlighted in bold the three conditions tested in Croes *et al.*[21]

a valid metabolic pathway. These assumptions are, however, not always justified. For example, the citric acid cycle has not been optimized to consist of the smallest possible number of enzymes but to produce energy and precursors for some metabolic pathways (e.g., amino acid biosynthesis). The second assumption does not hold for cyclic pathways or for pathways in which the same enzymes act repeatedly on a growing chain (e.g., fatty acid elongation). Consequently, these pathways can only be partly inferred.

Our pathfinding approach is also limited by the incomplete coverage of KEGG reactions by RPAIR (842 of 6580 reactions involved in small-molecule metabolism are not covered). In addition, RPAIRs are so far only annotated in KEGG and not cross-referenced to other major metabolic databases such as BioCyc[5] or Reactome,[41] so that our pathfinding approach can only be applied to KEGG/LIGAND metabolic data.

In general, metabolic pathfinding accuracy depends on the size and quality of the underlying metabolic network. If a reaction or compound is absent from the input network, a path containing this reaction or compound cannot be predicted. On the other side, a wrong connection between two compounds or reactions might lead to the prediction of an erroneous path. The KEGG networks evaluated in our study were not filtered (except to remove disconnected compounds, as well as non-small-molecule reactions and their reactants). Further filtering steps may therefore increase the accuracy of predicted paths.

The evaluation presented here is restricted to three organisms. It is worth wondering whether the approach is applicable to other organisms as well. In principle, our pathfinding approach is not organism-specific, but the metabolic network available might be biased for the model organisms that served to annotate the reference maps. In newly sequenced genomes, the pathways may differ from those reference pathways. However, if an organism employs alternative pathways by assembling in a different way the reactions and compounds covered by the current KEGG data, our method should be able to discover them. Of course, the ca 6000 reactions available in LIGAND are likely to miss some reactions that might be important for some organism-specific pathways or contain some annotation errors. In such case, our method will fail to assemble correctly the reactions and return incorrect pathways.

Another limitation of our metabolic pathfinding tool, but not of our approach in general, is its incapability to predict the direction of a pathway. The direction of a reaction is organism-specific (being dependent on the temperature and reactant concentrations in an organism). Because of this, we decided to treat all reactions as reversible in this evaluation study. However, the pathfinding algorithm can easily handle irreversible reactions, and the metabolic pathfinding tool allows the user to upload custom networks that may contain irreversible reactions.

Our compound weighting scheme exploits the fact that many compounds participating in metabolic pathways are only involved in a few reactions. Thus, it cannot deal well with pathways situated in the core of the metabolic network that consist of several highly connected compounds (i.e., glycolysis). Finding paths in the RPAIR graph alleviates this weakness, but some central pathways still escape detection.

Note that even though inferred pathways do not reproduce annotated pathways with 100% accuracy,

Table 3. Systematic pairwise comparison between experiments

Parameter	Superior value	Inferior value	Number of unequal comparisons	Number of times superior value has higher accuracy	Number of times inferior value has higher accuracy	Number of times both values have equal accuracy	Mean accuracy of superior value	Mean accuracy of inferior value	Difference between mean accuracies	Paired Wilcoxon signed-rank test *P*-value	Level of significance
Organisms merged											
Compound weight	Degree	Unit	52	52	0	0	0.77	0.63	0.14	1.80E−10	**
Graph type	RPAIR graph	Reaction-specific RPAIR graph	48	48	0	0	0.74	0.68	0.06	8.36E−10	**
Compounds filtered	True	False	50	31	19	2	0.73	0.66	0.07	2.30E−05	**
RPAIR class filtering	Main–trans	Main	32	25	7	0	0.72	0.69	0.03	8.85E−05	**
RPAIR class filtering	All	Main	32	25	7	0	0.72	0.69	0.03	0.000279793	**
Graph type	RPAIR graph	Reaction graph	8	8	0	0	0.74	0.53	0.21	0.004	*
Graph type	Reaction-specific RPAIR graph	Reaction graph	8	8	0	0	0.67	0.53	0.14	0.004	*
RPAIR weight	RPAIR class-specific	Unit	40	20	20	8	0.72	0.7	0.02	0.021	
Directed	False	True	17	12	5	35	0.7	0.7	0	0.138	
RPAIR class filtering	All	Main–trans	28	16	12	4	0.72	0.72	0	0.203	
Results for E. coli											
Compound weight	Degree	Unit	52	52	0	0	0.88	0.72	0.16	1.79E−10	**
Graph type	RPAIR graph	Reaction-specific RPAIR graph	48	48	0	0	0.85	0.79	0.06	8.33E−10	**
Compounds filtered	True	False	50	38	12	2	0.84	0.77	0.07	6.24E−07	**

RPAIR weight	RPAIR class-specific	Unit	33	24	9	15	0.83	0.81	0.02	0.000827796	**
Graph type	RPAIR graph	Reaction graph	8	8	0	0	0.83	0.62	0.21	0.004	*
Graph type	Reaction-specific RPAIR graph	Reaction graph	8	8	0	0	0.76	0.62	0.14	0.004	*
RPAIR class filtering	Main	All	32	23	9	0	0.83	0.81	0.02	0.025	
RPAIR class filtering	Main–trans	All	23	13	10	9	0.82	0.81	0.01	0.046	
RPAIR class filtering	Main	Main–trans	32	16	16	0	0.83	0.82	0.01	0.168	
Directed	False	True	11	6	5	41	0.8	0.8	0	0.518	
Results for S. cerevisiae											
Compound weight	Degree	Unit	52	48	4	0	0.6	0.47	0.13	3.19E−10	**
Compounds filtered	True	False	28	27	1	24	0.58	0.49	0.09	2.18E−06	**
Graph type	RPAIR graph	Reaction-specific RPAIR graph	48	40	8	0	0.55	0.53	0.02	8.05E−05	**
Graph type	RPAIR graph	Reaction graph	8	8	0	0	0.57	0.44	0.13	0.007	
Graph type	Reaction-specific RPAIR graph	Reaction graph	8	8	0	0	0.54	0.44	0.1	0.007	
RPAIR class filtering	Main–trans	Main	28	16	12	4	0.55	0.53	0.02	0.022	
RPAIR class filtering	All	Main–trans	28	18	10	4	0.55	0.55	0	0.023	
RPAIR weight	RPAIR class-specific	Unit	26	17	9	22	0.54	0.54	0	0.419	
Directed	False	True	6	3	3	46	0.53	0.53	0	0.663	
RPAIR class filtering	Main	All	30	16	14	2	0.53	0.55	0.02	0.967	

(continued on next page)

Table 3 (*continued*)

Parameter	Superior value	Inferior value	Number of unequal comparisons	Number of times superior value has higher accuracy	Number of times inferior value has higher accuracy	Number of times both values have equal accuracy	Mean accuracy of superior value	Mean accuracy of inferior value	Difference between mean accuracies	Paired Wilcoxon signed-rank test *P*-value	Level of significance
Results for H. sapiens											
Graph type	RPAIR graph	Reaction-specific RPAIR graph	48	45	3	0	0.62	0.54	0.08	1.41E−09	**
Compound weight	Degree	Unit	52	50	2	0	0.61	0.52	0.09	1.63E−09	**
RPAIR class filtering	Main–trans	Main	32	30	2	0	0.6	0.52	0.08	1.66E−06	**
Compounds filtered	True	False	30	28	2	22	0.58	0.54	0.04	1.75E−06	**
RPAIR class filtering	All	Main	32	30	2	0	0.61	0.52	0.09	2.39E−06	**
Graph type	RPAIR graph	Reaction graph	8	8	0	0	0.63	0.38	0.25	0.007	
Graph type	Reaction-specific RPAIR graph	Reaction graph	8	8	0	0	0.54	0.38	0.16	0.007	
RPAIR weight	RPAIR class-specific	Unit	20	12	8	28	0.58	0.57	0.01	0.017	
Directed FALSE	True		10	7	3	42	0.56	0.56	0	0.154	
RPAIR class filtering	All	Main–trans	12	8	4	20	0.61	0.6	0.01	0.155	

A Wilcoxon paired signed-rank test was applied to all pairs of parameter values, with all other parameters having the same value. This test computes Acc differences for combinations that differ by only one parameter value. *P*-value indicates the first error risk (i.e., the probability to consider a difference as significant when it is not). The table is sorted by ascending *P*-values, so the most significant parameter value differences appear on top.

* *P*-value below 0.005.

** *P*-value below 0.001.

they may be biochemically valid alternatives (e.g., aldosterone biosynthesis in humans). Our evaluation may thus underestimate the accuracy of our pathfinding approach.

Pathfinding and constraints

In the Introduction, we have discussed pathfinding approaches under the aspect of treatment of highly connected compounds. Another way to classify these approaches might rely on the constraints used to narrow down the huge number of possible solution pathways. We can list the following biologically or biochemically motivated constraints:

1. Pathway length/weight: The solution pathway should be as short (light) as possible (i.e., Croes *et al.*,[21] Arita,[28] Rahman *et al.*,[31] Sirava *et al.*,[42] and Beasley and Planes[43]).
2. Imposing or excluding nodes: Certain compounds or reactions should appear or should not appear in the solution pathway (e.g. Rahman *et al.*,[31] Sirava *et al.*,[42] and Mavrovouniotis[44]).
3. Mutual exclusion of reaction directions: The direct and reverse directions of a reaction should not appear together in a solution pathway.[21,44]
4. Maximal pathway length: The length (number of reactions) of the solution pathway should neither exceed a maximum nor fall below a minimum.[6,21]
5. Stoichiometric constraint: Stoichiometric balance should be respected between some or all of the compounds of the solution pathway (e.g. Beasley and Planes,[43] Mavrovouniotis,[44] and Seressiotis and Bailey[45]).

The first constraint is the most widely applied: All approaches based on the shortest pathfinding make use of it implicitly. Beasley and Planes employ pathway length as an objective function to be optimized (alternatively to ATP production maximization). Constraints 2–4 are already integrated in our pathfinding approach, except for imposing selected nodes, which we hope to deal with in the future by using multiseed pathfinding.

To our knowledge, Beasley and Planes presented the most recent approach capable of computing pathway stoichiometry. They used a strategy reminiscent of flux balance analysis that is based on the optimization of an objective function while satisfying a number of constraints. They evaluated this approach on a network reconstructed for *E. coli* consisting of 880 reactions and recovered, among others, the glycolysis pathway correctly. However, in a recent review, Planes and Beasley pointed out some disadvantages of the stoichiometric constraint.[46] First of all, it is unclear which compounds should be stoichiometrically balanced (internal compounds) and which can be left unconstrained (external compounds). Furthermore, they listed metabolic pathways in which compounds annotated to be internal are not balanced. In conclusion, they state that pathfinding without stoichiometric constraints is more suited for pathway analysis in genome-scale metabolic networks.

Perspectives

In the future, we hope to extend our method to more than two sets of input reactions or compounds. This will enable the inference of branched pathways and might allow for an increase in prediction accuracy when more input reactions or compounds are available. A complementary refinement of our method might be to select metabolic networks representing the metabolism of only one or several related organisms.

The compound weighting applied in this evaluation is based on a relatively simple rule (a compound weight is set equal to its degree in the metabolic network). In the future, alternative weighting schemes will be considered in order to achieve context-specific pathfinding. For instance, organism-specific pathfinding might be refined by applying weights to reactions according to their likelihood to be catalyzed in the organism of interest, rather than by selecting the subnetwork on the basis of genome annotations. Indeed, some enzyme-coding genes may have been missed in the annotation but play an essential role in some pathways. Another possibility is to weight reactions according to the level of expression of the corresponding enzymes, as measured by microarray experiments. A similar approach has been applied earlier by scoring paths according to the average correlation between the expression profiles of the corresponding genes.[47] Instead of scoring the paths *a posteriori*, weighted pathfinding would allow application of an *a priori* bias on pathfinding in order to favor reactions catalyzed by the products of up-regulated genes. Pathfinding could thus be tuned according to the physiological state of the cell under particular culture conditions.

Another perspective is to generalize the concept of pathfinding by applying constraint programming methods to the problem of metabolic pathway inference. In this context, Dooms et al. designed a general framework for constrained pathfinding in biochemical networks, named CP(BioNet).[48] At the time of writing, CP(BioNet) was restricted to networks consisting of, at most, 500 nodes, and we could thus not evaluate it on our metabolic networks, but its performance might be improved in the near future. The combination of this framework with the optimal conditions that we identified in the present study might result in a powerful and flexible metabolic pathfinding tool.

It would be of great interest to carry out a comparative evaluation of the different metabolic pathfinding approaches available. Although some comparative studies have been published,[38,46] a thorough evaluation either based on a community-based CASP-like blind protocol or performed by independent users is still missing.

Table 4. Pairwise comparison between parameter values, with each parameter being associated with its "best friends" (conditionally optimal combinations)

Parameter	Superior value	Inferior value	Number of unequal comparisons	Number of times superior value has higher accuracy	Number of times inferior value has higher accuracy	Number of times both values have equal accuracies	Mean accuracy of superior value	Mean accuracy of inferior value	Difference between mean accuracies	Paired Wilcoxon signed-rank test *P*-value	Unique best friends of superior value	Unique best friends of inferior value	Common best friends	Level of significance
Organisms merged														
Graph type	RPAIRs	Reaction graph	25	23	2	24	0.83	0.73	0.1	2.4E−05	RPAIRs; main–trans; Cpd unfiltered	Rct graph; all; directed	Rct unit weight; Cpd degree weight	**
Graph type	RPAIRs	Reaction-specific RPAIRs	24	23	1	25	0.83	0.77	0.06	4.7E−05	RPAIRs; main–trans	Rct-specific RPAIRs; all	Rct unit weight; Cpd degree weight; Cpd unfiltered	**
Graph type	Reaction-specific RPAIRs	Reaction graph	9	7	2	40	0.77	0.73	0.04	0.012	Rct-specific RPAIRs; Cpd unfiltered	Rct graph; directed	All; Rct unit weight; Cpd degree weight	
RPAIR class filtering	All	Main	10	9	1	39	0.83	0.79	0.04	0.033	All; Cpd unfiltered	Main; undirected; Cpd filtered	RPAIRs; Rct unit weight; Cpd degree weight	
RPAIR class filtering	Main–trans	Main	8	7	1	41	0.83	0.79	0.04	0.070	Main–trans; Cpd unfiltered	Main; undirected; Cpd filtered	RPAIRs; Rct unit weight; Cpd degree weight	
Compound weight	Degree	Unit	18	11	7	31	0.83	0.79	0.04	0.148	Main–trans; Rct unit weight; Cpd degree weight; Cpd unfiltered	All; RPAIR class-specific weight; Cpd unit weight; Cpd filtered	RPAIRs	
RPAIR weight	RPAIR class-specific	Unit weight	3	2	1	46	0.82	0.83	0.01	0.605	All; RPAIR class-specific weight	Main–trans; Rct unit weight	RPAIRs; Cpd degree weight; Cpd unfiltered	
RPAIR class filtering	All	Main–trans	4	2	2	45	0.83	0.83	0	0.821	All	Main–trans	RPAIRs; Rct unit weight; Cpd degree weight; Cpd unfiltered	
Directed	True	False	0	0	0	49	0.83	0.83	0	1	Directed	Undirected	RPAIRs; main–trans; Rct unit weight; Cpd degree weight; Cpd unfiltered	
Compounds filtered	True	False	0	0	0	49	0.83	0.83	0	1	Cpd filtered	Cpd unfiltered	RPAIRs; main–trans; Rct unit weight; Cpd degree weight	
Results for E. coli														
Graph type	RPAIRs	Reaction-specific RPAIRs	14	14	0	14	0.95	0.88	0.07	5.3E−04	RPAIRs; Cpd unfiltered	Rct-specific RPAIRs; main–trans; RPAIR class-specific weight; Undirected; Cpd filtered	Cpd degree weight	**
Graph type	RPAIRs	Reaction graph	14	14	0	14	0.95	0.84	0.11	5.3E−04	RPAIRs; Cpd unfiltered	Rct graph; all; Rct unit weight; directed	Cpd degree weight	**

RPAIR class filtering	Main–trans	Main	3	3	0	25	0.95	0.92	0.03	0.087	Main–trans	Main; Rct unit weight	RPAIRs; Cpd degree weight; Cpd unfiltered
RPAIR class filtering	All	Main	3	3	0	25	0.95	0.92	0.03	0.087	All; RPAIR class-specific weight	Main; Rct unit weight	RPAIRs; Cpd degree weight; Cpd unfiltered
Graph type	Reaction-specific RPAIRs	Reaction graph	3	3	0	25	0.88	0.84	0.04	0.091	Rct-specific RPAIRs; main–trans; RPAIR class-specific weight	Rct graph; all; Rct unit weight; directed	Cpd degree weight
Compound weight	Degree	Unit	6	4	2	22	0.95	0.91	0.04	0.417	Cpd degree weight; Cpd unfiltered	All; RPAIR class-specific weight; Cpd unit weight; Cpd filtered	RPAIRs
RPAIR class filtering	All	Main–trans	0	0	0	28	0.95	0.95	0	1	All; RPAIR class-specific weight	Main–trans	RPAIRs; Cpd degree weight; Cpd unfiltered
RPAIR weight	RPAIR class-specific	Unit weight	0	0	0	28	0.95	0.95	0	1	RPAIR class-specific weight	Main–trans; Rct unit weight	RPAIRs; Cpd degree weight; Cpd unfiltered
Directed	True	False	0	0	0	28	0.95	0.95	0	1	Directed	Undirected	RPAIRs; Cpd degree weight; Cpd unfiltered
Compounds filtered	True	False	0	0	0	28	0.95	0.95	0	1	Cpd filtered	Cpd filtered	RPAIRs; Cpd degree weight
			0	0	0	28	0.95	0.95	0	1	Cpd filtered	Cpd unfiltered	RPAIRs; Cpd degree weight
Results for S. cerevisiae													
Graph type	RPAIRs	Reaction graph	3	3	0	8	0.66	0.59	0.07	0.091	RPAIRs	Rct graph; directed	All; Rct unit weight; Cpd degree weight
Graph type	RPAIRs	Reaction-specific RPAIRs	3	3	0	8	0.66	0.62	0.04	0.091	RPAIRs; Rct unit weight	Rct-specific RPAIRs	All; Cpd degree weight
RPAIR weight	Unit weight	RPAIR class-specific	3	3	0	8	0.66	0.62	0.04	0.091	RPAIRs; Rct unit weight	Rct-specific RPAIRs; RPAIR class-specific weight	All; Cpd degree weight
RPAIR class filtering	All	Main	6	5	1	5	0.66	0.6	0.06	0.147	All; Rct unit weight	Main	RPAIRs; Cpd degree weight
Compound weight	Degree	Unit	6	5	1	5	0.66	0.57	0.09	0.147	All; Rct unit weight; Cpd degree weight	Cpd unit weight; Cpd filtered	RPAIRs
RPAIR class filtering	All	Main–trans	2	2	0	9	0.66	0.64	0.02	0.186	All	Main–trans	RPAIRs; Rct unit weight; Cpd degree weight
RPAIR class filtering	Main–trans	Main	4	3	1	7	0.64	0.6	0.04	0.290	Main–trans; Rct unit weight	Main	RPAIRs; Cpd degree weight
Graph type	Reaction-specific RPAIRs	Reaction graph	2	1	1	9	0.62	0.59	0.03	0.500	Rct-specific RPAIRs	Rct graph; Rct unit weight; directed	All; Cpd degree weight

(continued on next page)

Table 4 (*continued*)

Parameter	Superior value	Inferior value	Number of unequal comparisons	Number of times superior value has higher accuracy	Number of times inferior value has higher accuracy	Number of times both values have equal accuracies	Mean accuracy of superior value	Mean accuracy of inferior value	Difference between mean accuracies	Paired Wilcoxon signed-rank test *P*-value	Unique best friends of superior value	Unique best friends of inferior value	Common best friends	Level of significance
Directed	True	False	0	0	0	11	0.66	0.66	0	1	Directed	Undirected	RPAIRs; all; Rct unit weight; Cpd degree weight	
Compounds filtered	True	False	0	0	0	11	0.66	0.66	0	1	Cpd filtered	Cpd unfiltered	RPAIRs; all; Rct unit weight; Cpd degree weight	
Results for H. sapiens														
Graph type	RPAIRs	Reaction graph	8	7	1	2	0.7	0.59	0.11	0.021	RPAIRs	Rct graph; all; Rct unit weight; directed	Cpd degree weight	
Graph type	RPAIRs	Reaction-specific RPAIRs	8	7	1	2	0.7	0.64	0.06	0.040	RPAIRs	Rct-specific RPAIRs; all	Cpd degree weight	
RPAIR class filtering	Main–trans	Main	8	7	1	2	0.7	0.6	0.1	0.040	RPAIRs; main–trans	Rct-specific RPAIRs; main; Rct unit weight	Cpd degree weight	
RPAIR class filtering	All	Main	8	7	1	2	0.7	0.6	0.1	0.040	RPAIRs; all	Rct-specific RPAIRs; main; Rct unit weight	Cpd degree weight	
Graph type	Reaction-specific RPAIRs	Reaction graph	4	3	1	6	0.64	0.59	0.05	0.101	Rct-specific RPAIRs	Rct graph; Rct unit weight; directed	All; Cpd degree weight	
Compound weight	Degree	Unit	6	3	3	4	0.7	0.67	0.03	0.500	Cpd degree weight	RPAIR class-specific weight; Cpd unit	RPAIRs	
RPAIR class filtering	All	Main–trans	0	0	0	10	0.7	0.7	0	1	All	Weight; Cpd	RPAIRs; Cpd degree weight	
RPAIR weight	RPAIR class-specific	Unit weight	0	0	0	10	0.7	0.7	0	1	RPAIR class-specific weight	Rct unit weight	RPAIRs; Cpd degree weight	
Directed	True	False	0	0	0	10	0.7	0.7	0	1	Directed	Undirected	RPAIRs; Cpd degree weight	
Compounds filtered	True	False	0	0	0	10	0.7	0.7	0	1	Cpd filtered	Cpd unfiltered	RPAIRs; Cpd degree weight	

For each parameter, we selected the combination of other parameters giving the best accuracy. Wilcoxon test was then performed by counting the number of pathways for each alternative value, giving a better, worse, or equal accuracy, respectively.

Cpd: compound; Rct: reaction.

* *P*-value below 0.005.

** *P*-value below 0.001.

Materials and Methods

Network construction

We constructed three different networks from KEGG/LIGAND (release 41.0).

The first graph, named *reaction graph*, was built from all reactions and compounds present in KEGG/LIGAND. The resulting graph is composed of 6359 reactions and 5312 compounds, which are connected by 26,786 edges.

The second graph, termed *RPAIR graph*, was constructed from all reactant pairs (7058) in KEGG RPAIR (release 41.0) and all compounds involved in at least one of these reactant pairs (4297). It has fewer compounds than the reaction graph because RPAIR does not cover all the reactions and compounds listed in LIGAND.

Finally, we constructed a reaction-specific RPAIR graph, where each reaction is divided into its reactant pairs. In this graph, the same RPAIR can correspond to distinct nodes if it is part of multiple reactions. This graph contains 12,828 reaction-specific reactant pairs and as many compounds as the RPAIR graph (4297).

All three networks (or graphs, to use the mathematical term) are bipartite and not organism-specific. We excluded glycans and orphan nodes (nodes not connected to any other node), as well as reactions having a substrate and a product with an identical identifier in KEGG (these correspond to polymerization reactions). The reaction/RPAIR and compound node numbers given above refer to these filtered networks. The KEGG/LIGAND database contains a number of problematic entries (see Poolman *et al.*,[49] Félix and Valiente,[50] and Ott and Vriend[51]; e.g., generic compounds, duplicated reactions, and unbalanced reactions). However, further filtering steps are beyond the scope of this evaluation.

Mutual exclusion between reaction nodes

As discussed in Croes *et al.*,[37] the direction of a reaction (or a reaction pair) depends on physiological conditions in an organism (substrate concentration, product concentrations, and temperature). In our previous work, we thus considered that each reaction can be traversed either in forward direction or in reverse direction. Each reaction was therefore represented as a pair of nodes for the forward direction and the reverse direction, respectively. To prevent the pathfinding algorithm from traversing the same reaction/reactant pair twice, the forward and reverse directions exclude each other mutually. In the RPAIR and reaction-specific RPAIR graphs, we ensure, in addition, that reactant pairs belonging to the same reaction exclude each other.

Parameters

Table 1 summarizes the different parameters used for graph construction. In total, the values of the parameters listed (graph construction, graph structure, compound and reaction weights, and compound and reaction filtering) can be combined in 104 possible ways.

Reference pathways

In order to measure the accuracy of pathfinding in metabolic graphs, a set of reference pathways is needed. We chose the metabolic pathways stored in the aMAZE database[52] (version 2006) because they have been carefully annotated and provide side/main compound classification and cross-reference reactions and compounds to KEGG/LIGAND. Since pathfinding cannot deal with branched pathways, we selected the linear segments as in our previous work[37] and removed side compounds. Pathways composed of less than three reactions were discarded, since finding them would be trivial. For the two RPAIR graphs, reactions of the reference pathways had to be mapped to their corresponding reactant pairs. Each reaction was replaced by its main reactant pair. When a reaction was associated with several main reactant pairs, we selected only those whose substrate and product were part of the linearized pathway. For three of the *E. coli* pathways (alanine biosynthesis and the two branches of methionine biosynthesis), mapping to main reactant pairs was not sufficient to obtain the fully connected pathway, and we had to use trans reactant pairs.

The aMAZE database contained 116 pathways: 55 from *E. coli*, 29 from *S. cerevisiae*, and 32 from *H. sapiens*. Of these, 7 were cyclic (*E. coli*, 2; *S. cerevisiae*, 4; *H. sapiens*, 1), 25 were branched (*E. coli*, 13; *S. cerevisiae*, 5; *H. sapiens*, 7), and 46 contained less than three reactions (*E. coli*, 20; *S. cerevisiae*, 14; *H. sapiens*, 12). After filtering and linearization, 69 pathways remained (*E. coli*, 37; *S. cerevisiae*, 14; *H. sapiens*, 18). After mapping of reactions to main/trans RPAIRs, we ended up with the 55 pathways used for evaluation (*E. coli*, 32; *S. cerevisiae*, 11; *H. sapiens*, 12).

Metabolic pathfinding algorithm

We find paths in weighted graphs using the *k*-shortest paths algorithm developed and implemented by Jimenez and Marzal.[53] By introducing pseudo nodes (a graph transformation described in Duin *et al.*[54]), we enable the search between a set of start nodes and a set of end nodes. Starts and ends can be compounds, reactions, or both. For given sets of start and end nodes, we regard as an inferred pathway the union of all paths of first rank (having equal weights) between them. Thus, inferred pathways can contain alternative branches.

Evaluation procedure

The evaluation procedure consists in finding the *k*-shortest (or lightest) paths between the start and the end reactions of a reference pathway and comparing the intermediate nodes of the inferred path with those of the annotated pathway. The motivation for searching paths from reaction to reaction (rather than from compound to compound) is that this is more relevant to metabolic reconstruction, where one starts from sets of enzyme-coding genes.

To compute the accuracy of an inferred pathway, we consider the intersection of node sets of the inferred and reference pathways. Nodes appearing in both pathways count as true positives, whereas nodes absent in the inferred pathway but present in the reference pathway are false negatives. Nodes found in the inferred pathway but not in the reference pathway are false positives. Since start and end nodes are known beforehand, they are not considered for those matching statistics.

Note that, in our previous publication,[21] alternative reactions linking a pair of compounds were not counted as false positives. In the present study, we applied more stringent criteria in order to precisely quantify the impact of the RPAIR annotations. For the sake of comparison, we also applied all the tests with the previous scoring scheme, and the results are available as Tables S5 and S6.

Sn is the fraction of annotated nodes that are found in the inferred pathway:

$$\mathrm{Sn} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$$

PPV is defined as the fraction of inferred nodes that belong to the annotated pathway:

$$\mathrm{PPV} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$$

Given both Sn and PPV, we calculate the accuracy of an inferred pathway as their geometric mean:

$$\mathrm{Acc} = \mathrm{sqrt}(\mathrm{SN} \times \mathrm{PPV})$$

The evaluation of a specific parameter combination proceeds as follows: First, we identify the start and end reactions (or main reactant pairs) of one reference pathway. Metabolic pathfinding is run on these input reactions in one of the three metabolic graphs and returns a pathway. By comparing the predicted pathway with the reference pathway, the accuracy of pathway prediction is calculated. These steps are performed for each pathway in the reference set and for each possible combination of the parameters described above. Since we deal with 104 parameter combinations, 104 such evaluations have been performed.

Wilcoxon paired signed-rank test

Since we perform a number of experiments with different parameter values, we need a statistical test to measure the significance of accuracy differences that result from having used two different values for the same parameter. We have chosen the Wilcoxon paired signed-rank test for this task because it makes no assumptions on sample distribution.

Let us consider one group of pathfinding experiments performed with one value and another group of experiments performed with another value for the same parameter. The null hypothesis states that there is no accuracy difference between the two experiment groups. The alternative hypothesis says that one of the two experiment groups yields pathfinding accuracies that belong to another (right-shifted) distribution compared to those obtained for the other experiment group, and that, consequently, one parameter value improves pathfinding accuracy as compared to the other to a certain level of significance.

For the experimentwise Wilcoxon test, we paired all experiments that differed only by the parameter in question. We thus obtained two vectors, where each entry represents the pathfinding accuracy averaged over all paths of an experiment. The significance of pathfinding accuracy difference is then computed on these paired vectors.

In contrast, for the best-combination-wise Wilcoxon test, we paired only those experiments that yielded the highest accuracies for the two values of a parameter in question. We then calculated the significance of the accuracy difference by comparing the pathfinding accuracies of these experiments pathwaywise.

For both Wilcoxon tests, we used the function wilcox.test in R to compute *P*-values.

Implementation and availability

The metabolic pathfinding tool† is available as part of Network Analysis Tools.[39] It allows the enumeration of the shortest paths in the three graph types and lets the user choose between different combinations of compound and reaction weights. The optimal parameter values obtained by our evaluation are set as default. KEGG identifiers of compounds, reactions and reactant pairs, EC numbers, and compound names can be given to specify start and end nodes. In addition, multiple start and end nodes can be provided. It is also possible to find paths in organism-specific metabolic networks extracted from KEGG.

The complete results of our evaluation (104 experiments for *E. coli*, *S. cerevisiae*, and *H. sapiens*) can be found at the RSAT Web site‡.

Acknowledgements

We would like to thank Jean-Noël Monette and Pierre Schaus for making available useful modifications to the REA algorithm. They also implemented a Python wrapper around REA that inspired the Java wrapper employed for pathfinding evaluation.

We acknowledge the aMAZE team for their support during the initial stages of this work. We used their Java graph library and, in addition, one of them (Olivier Hubaut) suggested the graph transformation that extended pathfinding with two nodes to pathfinding with two sets of nodes. Thanks as well to Pierre Dupont and Grégoire Dooms for helpful discussions.

K.F. was supported by Actions de Recherches Concertées de la Communauté Française de Belgique (ARC grant no. 04/09-307). The laboratory is a member of the BioSapiens Network of Excellence funded under the Sixth Framework Program of the European Communities (LSHG-CT-2003-503265). The work was also supported by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, under project P6/25 (BioMaGNet).

Supplementary Data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.jmb.2009.03.006

References

1. Berg, J. M., Tymoczko, J. L. & Stryer, L. (2002). In *Biochemistry*, 5th edit. (Stryer, L., ed.), W. H. Freeman and Company, New York.

† http://rsat.ulb.ac.be/neat/

‡ http://rsat.ulb.ac.be/pathfindingsupplementref/index.html

2. Goto, S., Bono, H., Ogata, H., Fujibuchi, W., Nishioka, T., Sato, K. & Kanehisa, M. (1997). Organizing and computing metabolic pathway data in terms of binary relations. *Pac. Symp. Biocomput.* 175–186.
3. Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M. *et al.* (2008). KEGG for linking genomes to life and the environment. *Nucleic Acids Res.* **36**, D480–D484.
4. Karp, P. D., Riley, M., Paley, S. & Pellegrini-Toole, A. (1996). EcoCyc: an encyclopedia of *Escherichia coli* genes and metabolism. *Nucleic Acids Res.* **24**, 32–39.
5. Caspi, R., Foerster, H., Fulcher, C. A., Kaipa, P., Krummenacker, M., Latendresse, M. *et al.* (2008). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.* **36**, D623–D631.
6. Kueffner, R., Zimmer, R. & Lengauer, T. (2000). Pathway analysis in metabolic databases via differential metabolic display. *Bioinformatics*, **16**, 825–836.
7. Förster, J., Famili, I., Fu, P., Palsson, B.Ø. & Nielsen, J. (2003). Genome-scale reconstruction of the *Saccharomyces cerevisiae* metabolic network. *Genome Res.* **13**, 244–253.
8. Duarte, N. C., Becker, S. A., Jamshidi, N., Thiele, I., Mo, M. L., Vo, T. D. *et al.* (2007). Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proc. Natl Aced. Sci.* **104**, 1777–1782.
9. Gaasterland, T. & Selkov, E. (1995). Reconstruction of metabolic networks using incomplete information. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **3**, 127.
10. Arita, M. (2000). Metabolic reconstruction using shortest paths. *Simul. Pract. Theory*, **8**, 109–125.
11. Ma, H. & Zeng, A. -P. (2003). Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics*, **19**, 270–277.
12. Karp, P. D., Ouzounis, C. A., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahrén, D. *et al.* (2005). Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Res.* **33**, 6083–6089.
13. Moriya, Y., Itoh, M., Okuda, S., Yoshizawa, A. C. & Kanehisa, M. (2007). KAAS: an automatic genome annotation and pathway reconstruction server. *Nucleic Acids Res.* **35**, W182–W185.
14. DeJongh, M., Formsma, K., Boillot, P., Gould, J., Rycenga, M. & Best, A. (2007). Toward the automated generation of genome-scale metabolic networks in the SEED. *BMC Bioinf.* **8**, 139.
15. Paley, S. M. & Karp, P. D. (2002). Evaluation of computational metabolic-pathway predictions for *Helicobacter pylori*. *Bioinformatics*, **18**, 705–714.
16. Kharchenko, P., Vitkup, D. & Church, G. M. (2004). Filling gaps in a metabolic network using expression information. *Bioinformatics*, **20**, i178–i185.
17. Kharchenko, P., Chen, L., Freund, Y., Vitkup, D. & Church, G. M. (2006). Identifying metabolic enzymes with multiple types of association evidence. *BMC Bioinf.* **7**, 177.
18. Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. & Barabási, A. -L. (2000). The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.
19. Fell, D. A. & Wagner, A. (2000). The small world of metabolism. *Metab. Eng.* **18**, 1121–1122; Nature America, Inc.
20. Arita, M. (2004). The metabolic world of *Escherichia coli* is not small. *Proc. Natl Acad. Sci.* **101**, 1543–1547.
21. Croes, D., Couche, F., Wodak, S. & van Helden, J. (2006). Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.* **356**, 222–236.
22. van Helden, J., Wernisch, L., Gilbert, D. & Wodak, S. (2002). Graph-based analysis of metabolic networks. *Ernst Schering Res. Found. Workshop*, **38**, 245–274.
23. Schuster, S., Dandekar, T. & Fell, D. A. (1999). Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *TIBTECH*, **17**, 53–60.
24. Teusink, B., Wiersma, A., Molenaar, D., Francke, C., de Vos, W. M., Siezen, R. J. & Smid, E. J. (2006). Analysis of growth of *Lactobacillus plantarum* WCFS1 on a complex medium using a genome-scale metabolic model. *J. Biol. Chem.* **281**, 40041–40048.
25. Edwards, J. S. & Palsson, B.Ø. (2001). Metabolic flux balance analysis and the *in silico* analysis of *Escherichia coli* K-12 gene deletions. *BMC Bioinf.* **1**, 1.
26. Karp, P. D. & Paley, S. M. (1994). Representations of metabolic knowledge: pathways. *Proc. 2nd Int. Conf. Intell. Syst. Mol. Biol.* 203–211.
27. Arita, M., Asai, K. & Nishioka, T. (2000). Graph modeling of metabolism. *J. Jpn. Soc. Artif. Intell.* **15**, 703–710.
28. Arita, M. (2003). *In silico* atomic tracing by substrate–product relationships in *Escherichia coli* intermediary metabolism. *Genome Res.* **13**, 2455–2466.
29. McShan, D. C., Rao, S. & Shah, I. (2003). PathMiner: predicting metabolic pathways by heuristic search. *Bioinformatics*, **19**, 1692–1698.
30. Boyer, F. & Viari, A. (2003). *Ab initio* reconstruction of metabolic pathways. *Bioinformatics*, **19**, ii26–ii34.
31. Rahman, S. A., Advani, P., Schunk, R., Schrader, R. & Schomburg, D. (2004). Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). *Bioinformatics*, **21**, 1189–1193.
32. Blum, T. & Kolhbalcher, O. (2008). Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks. *J. Comput. Biol.* **15**, 565–576.
33. Hou, B. K., Ellis, L. B. M. & Wackett, L. P. (2004). Encoding microbial metabolic logic: predicting biodegradation. *J. Ind. Microbiol. Biotechnol.* **31**, 261–272.
34. Kotera, M., Hattori, M., Oh, M. -A., Yamamoto, R., Komeno, T., Yabuzaki, J. *et al.* (2004). RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions. *Genome Inf.* **15**, P062.
35. Kotera, M., Okuno, Y., Hattori, M., Goto, S. & Kanehisa, M. (2004). Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. *J. Am. Chem. Soc.* **126**, 16487–16498.
36. Oh, M., Yamada, T., Hattori, M., Goto, S. & Kanehisa, M. (2007). Systematic analysis of enzyme-catalyzed reaction patterns and prediction of microbial biodegradation pathways. *J. Chem. Inf. Model*, **47**, 1702–1712.
37. Croes, D., Couche, F., Wodak, S. & van Helden, J. (2005). Metabolic PathFinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Res.* **33**, W326–W330.
38. de Figueiredo, L. F., Schuster, S., Kaleta, C. & Fell, D. A. (2008). Can sugars be produced from fatty acids? A test case for pathway analysis tools. *Bioinformatics*, **24**, 2615–2621.
39. Brohee, S., Faust, K., Lima-Mendez, G., Sand, O., Janky, R., Vanderstocken, G. *et al.* (2008). NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Res.* **36**.
40. Pasqualini, J. R. (2005). Enzymes involved in the formation and transformation of steroid hormones in

the fetal and placental compartments. *J. Steroid Biochem. Mol. Biol.* **97**, 401–415.

41. Vastrik, I., D'Eustachio, P., Schmidt, E., Joshi-Tope, G., Gopinath, G., Croft, D. *et al.* (2007). Reactome: a knowledge base of biologic pathways and processes. *Genome Biol.* **8**, R39.
42. Sirava, M., Schaefer, T., Eiglsperger, M., Kaufmann, M., Kohlbacher, O., Bornberg-Bauer, E. & Lenhof, H. P. (2002). BioMiners—modeling, analyzing, and visualizing biochemical pathways and networks. *Bioinformatics*, **18**, S219–S230.
43. Beasley, J. E. & Planes, F. J. (2007). Recovering metabolic pathways via optimization. *Bioinformatics*, **23**, 92–98.
44. Mavrovouniotis, M. L. (1993). Identification of qualitatively feasible metabolic pathways. In *Artificial Intelligence and Molecular Biology*, pp. 325–364, American Association for Artificial Intelligence, Menlo Park, CA.
45. Seressiotis, A. & Bailey, J. E. (1986). MPS: an algorithm and data base for metabolic pathway synthesis. *Biotechnol. Lett.* **8**, 837–842.
46. Planes, F. J. & Beasley, J. E. (2008). A critical examination of stoichiometric and path-finding approaches to metabolic pathways. *Brief. Bioinf.* **9**, 422–436.
47. Zien, A., Kuffner, R., Zimmer, R. & Lengauer, T. (2000). Analysis of gene expression data with pathway scores. *ISMB*, **8**, 407–417.
48. Dooms, G., Deville, Y. & Dupont, P. (2004). Constrained path finding in biochemical networks. In *5èmes Journées Ouvertes Biologie Informatique Mathématiques, JOBIM 2004*.
49. Poolman, M. G., Bonde, B. K., Gevorgyan, A., Patel, H. H. & Fell, D. A. (2006). Challenges to be faced in the reconstruction of metabolic networks from public databases. *IEE Proc. Syst. Biol.* **153**, 379–384.
50. Félix, L. & Valiente, G. (2007). Validation of metabolic pathway databases based on chemical substructure search. *Biomol. Eng.* **24**, 327–335.
51. Ott, M. A. & Vriend, G. (2006). Correcting ligands, metabolites, and pathways. *BMC Bioinf.* **7**, 517.
52. Lemer, C., Antezana, E., Couche, F., Fays, F., Santolaria, X., Janky, R. *et al.* (2004). The aMAZE LightBench: a web interface to a relational database of cellular processes. *Nucleic Acids Res.* **32**, D443–D448.
53. Jimenez, V. M. & Marzal, A. (1999). Computing the K Shortest Paths: A New Algorithm and an Experimental Comparison. *Proc. 3rd Int. Workshop Algorithm Eng. (WAE 1999), Lecture Notes in Computer Science*, vol. 1668, pp. 15–29 Springer-Verlag, London.
54. Duin, C. W., Volgenant, A. & Vo, S. (2004). Solving group Steiner problems as Steiner problems. *Eur. J. Oper. Res.* **154**, 323–329.

3 Multiple-end metabolic pathway prediction

Presented article:
K. Faust, P. Dupont, J. Callut and J. van Helden
Pathway discovery in metabolic networks by subgraph extraction
Submitted.

3.1 Introduction

After having established a strategy to increase the accuracy of two-end path finding, the next step of the thesis was to develop, evaluate and apply multiple-end pathway prediction approaches. In contrast to path finding, multiple-seed pathway prediction can predict pathways starting from or ending in several compounds or reactions. In addition, the ability to treat more than two seeds allows to predict metabolic pathways from groups of enzyme-coding genes, enzymes, reactions or compounds. This in turn can serve the interpretation of high-throughput data sets featuring these seed groups.

This chapter presents the development and evaluation of multiple-end pathway prediction approaches, whereas the next chapter (4) presents their application to a microarray data set.

Seven algorithms were evaluated, one of them based on random walks (kWalks [48]), and three others on shortest paths. The other three algorithms are combinations of the shortest-paths based algorithms with kWalks. The algorithms are presented in section 9.1. All algorithms rely on the extraction of a subgraph from the input network, which represents the predicted pathway.

3.2 Contribution

P. Dupont and J. Callut developed and implemented the kWalks algorithm. K. Faust implemented the other algorithms (except for Klein-Ravi), carried out the evaluation and wrote the article. P. Dupont and J. van Helden revised the article.

3.3 Methods

The evaluation of the algorithms was carried out on a network constructed from MetaCyc, with 71 reference pathways taken from *S. cerevisiae* (see section 9.2 for the pathway list). In contrast to the evaluation of path finding presented in chapter 2, a linearization of the reference pathways was unnecessary, since the evaluation of multiple-end pathway prediction should also quantify the prediction accuracy for branched pathways.

Each of the seven algorithms was launched several times on each of the 71 pathways, to predict pathways for increasing number of seed reactions.

In addition, the performance of the algorithms was evaluated for various parameters, e.g. network directionality, different weight policies, kWalks iteration number and others. KWalks outputs node and edge relevances, which can serve as new node or edge weights. These weights can be refined by calling kWalks iteratively on the input network, whose weights are up-dated in each round with the node/edge relevances computed by kWalks. The kWalks iteration number refers to the number of times kWalks is repeated.

3.4 Results

The algorithm reaching highest accuracy ($\sim$77%) was a hybrid of kWalks and Takahashi-Matsuyama [154]. The hybrid algorithm takes advantage of the complementary strengths of kWalks (high sensitivity, low computational complexity) and shortest-paths based approaches (high PPV).

The evaluation also yielded the following insights:

- KWalks can be used to discover weights in case a good weight policy is not at hand.
- Combining a shortest-paths based algorithm with kWalks increases its speed.
- Iterations of more than three do not increase the accuracy of kWalks any further.
- Reduction of the intermediate network size (i.e. its node number) in the hybrid approach down to a certain percentage (0.5% in this evaluation) increases the prediction accuracy.

3.5 Conclusion

A combination of a random walk-based (kWalks) with a shortest-paths based (Takahashi-Matsuyama) approach yields a pathway prediction accuracy of $\sim$77% in the weighted MetaCyc network, which was the highest achieved for any algorithm or parameter combination.

Pathway discovery in metabolic networks by subgraph extraction

Karoline Faust [1,*], Pierre Dupont [2,*], Jérôme Callut [2] and Jacques van Helden [1,*]

[1]Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe), Université Libre de Bruxelles, Campus Plaine - CP263, Boulevard du Triomphe, 1050 Bruxelles, Belgium.
[2] UCL Machine Learning Group, Computing Science and Engineering Department, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Subgraph extraction is a powerful technique to predict pathways from biological networks and a set of query items (e.g. genes, proteins, compounds...). It can be applied to a variety of different data types, such as gene expression, protein levels, operons or phylogenetic profiles. In this article, we investigate different approaches to extract relevant pathways from metabolic networks. Although these approaches have been adapted to metabolic networks, they are generic enough to be adjusted to other biological networks as well.

Results: We comparatively evaluated seven sub-network extraction approaches on 71 known metabolic pathways from *S. cerevisiae* and a metabolic network obtained from MetaCyc. The best performing approach is a novel hybrid strategy, which combines a random walk-based reduction of the graph with a shortest-paths based algorithm, and which recovers the reference pathways with an accuracy of $\sim$ 77%.

Availability: Most of the presented algorithms are available as part of the network analysis tool set (NeAT). The kWalks method is released under the GPL3 license.

Contact: kfaustulb.ac.be

1 INTRODUCTION

Pathway inference aims to extract a meaningful pathway given a biological network (e.g. protein-protein interaction or metabolic network) and a set of query items (e.g. genes, proteins, compounds). This methodology may serve to predict pathways from a variety of data types, such as gene expression, operons, phylogenetic profiles or protein levels.

To our knowledge, Zien *et al.* (2000) were the first to infer pathways from a biological network. They construct a bipartite metabolic network consisting of compound and reaction nodes, where enzyme-coding genes are linked to reactions via their EC numbers, and subsequently enumerate all possible paths between a source node (D-glucose) and a target node (pyruvate) under certain constraints. The score of each path is computed on the basis of expression values of the genes involved in this path as measured with microarrays. This method ranks predicted paths according to their degree of up- or down-regulation.

Ideker *et al.* (2002) extended this idea to the extraction of more complex, non-linear sub-networks in protein-protein and protein-DNA networks given yeast gene expression data. Sub-networks are considered active whenever they involve highly expressed genes. Such sub-networks can be identified by sampling the space of possible sub-networks with simulated annealing. The authors also mention several strengths of the sub-network extraction approach as compared to traditional gene clustering, for instance the consideration of genes that are only weakly differentially expressed.

Scott *et al.* (2005) also search for sub-networks in protein-protein and protein-DNA interaction networks given gene expression data. To our knowledge, their algorithm is the first to tackle the Steiner tree problem (Hwang *et al.* (1992)) on biological networks in order to connect nodes of interest (i.e. differentially expressed genes).

Rajagopalan and Agarwal (2005) integrate various data sources (TransFac, HumanCyc and Ingenuity Pathways Knowledge Base) into a network of gene-metabolite relationships. Query nodes in this network are connected by an algorithm based on breadth-first search. A key contribution of these authors is the systematic evaluation of their subgraph extraction approach on both simulated data and known pathways taken from BioCarta.

Noirel *et al.* (2008) apply sub-network extraction to proteomics data (that is enzyme level ratios, measured by mass spectrometry) from the cyanobacterium *Nostoc*. A sub-network is extracted from a weighted KEGG metabolic network by generating paths around each up-regulated enzyme node up to a given maximal weight and subsequently filtering these paths according to the number of up-regulated enzymes contained in them. The filtered paths are then merged to form a network whose connected components are considered as the extracted sub-networks.

Dittrich *et al.* (2008) identify high-scoring sub-networks in protein-protein interaction networks with a strategy similar to Scott *et al.* (2005), by applying an algorithm that solves the Steiner tree problem exactly. Interestingly, their method allows to report sub-optimal solutions with a user-specified distance to previously listed solutions. The pathway prediction approach is validated on simulated data.

[*]to whom correspondence should be addressed

Antonov and co-workers predict metabolic pathways from KEGG data and from input genes mapped to reactions (Antonov *et al.* (2008)) or input compounds (Antonov *et al.* (2009)). Seed nodes separated by one edge are added to a growing sub-network, which may consist of several components. The component covering most seeds is considered as the inferred pathway. The procedure is repeated for distances of 2,3... edges, resulting in a set of distance-specific predictions. This sub-network extraction procedure is available via two web tools specific to metabolic data.

In this article, we present a systematic assessment of sub-network extraction accuracy given a metabolic network. We evaluate the performance of four different algorithms (combined in seven approaches) on the basis of 71 pathways obtained from MetaCyc. One of these algorithms (pair-wise K-shortest paths) has been developed for this study and two other algorithms (Takahashi-Matsuyama, kWalks) have apparently not yet been applied to sub-network extraction in biological networks. The extraction techniques considered here are not specific to metabolic networks or gene expression data. They can be applied to any biological network and to any data set generating specific nodes of interest (e.g. functionally related groups of genes/enzymes as derived from phylogenetic co-occurrence, operons, gene fusion events etc.).

2 METHODS

2.1 Metabolic network construction

In order to predict metabolic pathways, we need to represent metabolic data as a network (or a graph, to use the mathematical term). We selected MetaCyc (Krieger *et al.* (2004)), the well-curated tier of BioCyc (Caspi *et al.* (2008)), as our data source, and constructed a bipartite, directed graph from all small molecule entries and their associated reactions contained in the OWL file of MetaCyc (Release 11.0). The resulting graph consists of 4,891 compound nodes and 5,358 reaction nodes. As discussed in Croes *et al.* (2005), reactions that are annotated as irreversible can be reversed depending on physiological conditions (substrate and product concentrations, temperature). Consequently, we represent each reaction as a pair of nodes, for the forward and the reverse directions, respectively. To prevent the paths-based algorithms to cross the same reaction twice, forward and reverse direction are mutually exclusive. After this duplication of reaction nodes, we obtain a directed network with 15,607 nodes and 43,938 edges, referred hereafter as the MetaCyc network.

We constructed two variants of the MetaCyc network: the directed one described above and an undirected network, in which reaction nodes are not duplicated. In both cases however, the weight matrix is designed to be symmetric.

2.2 Weight policies

Metabolic networks contain hub compounds such as H_2O, NADP and ATP, which are involved in a large number of reactions. A straightforward graph traversal algorithm would preferentially cross these compounds, resulting in biochemically invalid paths that connect for instance D-glucose with pyruvate in one reaction step via ADP. Available path finding tools (which extract a sub-network from a single source and target node) try to solve this problem by considering compound structure (e.g. Arita (2000); McShan *et al.* (2003); Rahman *et al.* (2004); Blum and Kohlbacher (2008)), network weights (Croes *et al.* (2005, 2006); Blum and Kohlbacher (2008)), annotated reactant pairs (Faust *et al.* (2009)) or rules (Ellis *et al.* (2008)). We adopted the weighting approach and tested three different weight policies. The simplest one (“unit weight”) sets all node weights to one. The second policy (“compound degree weight”) penalizes highly connected compounds by assigning to each compound a weight equal to its degree, whilst setting to each reaction a weight of one. The third weight policy (“inflated compound degree weight”) takes to the power of two the node weights defined by the second weight policy. The purpose is to enlarge weight differences between highly and weakly connected compound nodes. For most algorithms, the node-weighted network had to be converted to an edge/arc-weighted network, by taking for each edge/arc the mean of weights of its two adjacent nodes.

2.3 Reference pathways

We obtained a selected set of 71 known *S. cerevisiae* pathways from MetaCyc (Release 11.0). All pathways in this reference set consist of at least 5 nodes and are included in the largest connected component of the MetaCyc network. On average, the pathways are composed of 13 nodes and in addition, more than half of them are branched and/or cyclic.

2.4 Algorithms

We measured the sub-network extraction accuracy of four different algorithms: three are based on shortest paths (Takahashi and Matsuyama (1980); Klein and Ravi (1995), pair-wise K-shortest paths) and one is based on random walks (Dupont *et al.* (2006); Callut (2007)). In addition, we combined the random walks-based approach with each of the three shortest paths-based approaches, thus testing altogether seven approaches.

2.4.1 Common features of the extraction algorithms All algorithms extract sub-networks by connecting a set of selected nodes (the seed nodes) in the input network. The problem of connecting seed nodes in a weighted network such that the weight of the resulting sub-network is minimized is an instance of the Steiner tree problem which is known to be NP-complete (Karp (1972)). The Takahashi-Matsuyama, the Klein-Ravi and pair-wise K-shortest paths algorithms tackle the Steiner tree problem approximately using different heuristics.

The kWalks approach takes a qualitatively different approach to subgraph extraction by efficiently computing the set of edges most likely to be used while walking from a seed node to any other one. The weights in the network obviously influence the random walks together with the network topology.

2.4.2 Challenges faced by metabolic pathway inference algorithms The metabolic pathway inference algorithms face the following challenges.

(1) Be able to cope with weighted networks.

(2) Allow the input graph to be directed. In undirected graphs the paths-based approaches would not make the difference between reaction products and substrates, and would thus establish artefactual links from substrate to substrate, or from product to product. This requirement is not met by the implementation of Klein-Ravi used for evaluation.

(3) Treat forward and reverse direction of reactions as mutually exclusive. Without mutual exclusion of forward and reverse reaction direction nodes, the same reaction may appear twice in a shortest path. The kWalks method does not distinguish between forward and reverse reactions because it is not based on the explicit computation of paths.

(4) Be able to process seed node groups instead of seed nodes. The reaction mechanism(s) of an enzyme is (are) usually described by its EC number(s). But this annotation is ambiguous, because reactions with the same EC number may differ by their co-factor or by their substrate. For instance, homoserine dehydrogenase with EC number 1.1.1.3 converts L-homoserine into L-aspartate 4-semialdehyde. There are two reactions associated to this EC number (having either NAD+ or NADP+ as a co-factor), but only one of these may actually occur in the pathway to be inferred. An algorithm handling seed node groups can treat all reactions of EC number 1.1.1.3 as belonging to the same group. As soon as one of the group members is connected to the sub-network, the seed node group is considered to be connected as well. To address this last requirement, we applied the graph transformation suggested by Duin *et al.* (2004). The idea is to introduce pseudo nodes, which connect all members of a seed node group in the input graph. Thus, when we mention seed nodes, these nodes may be artificial

nodes that represent a group of seeds considered as equivalent, and from which only one has to be included in the result.

Each algorithm takes as input the graph, the seed nodes and a weight policy. kWalks requires additional parameters discussed in section 2.4.6.

We will first discuss the shortest paths-based approaches. Except for Klein-Ravi, they rely on the REA algorithm (Jimenez and Marzal (1999)) to compute K-shortest paths. REA enumerates all paths between a start and an end node in the order of their length. In a weighted graph, paths are listed in the order of their weight. Note that according to the definition of a path, a node can occur only once in the path. The value of K is dynamically set such that all paths of minimal weight are collected. The paths returned by REA are filtered to avoid paths containing mutually exclusive nodes.

The computational complexities of all algorithms described below are expressed in terms of n and m, respectively the number of nodes and edges in the input graph, as well as s, the number of seed nodes.

2.4.3 Klein-Ravi The algorithm by Klein and Ravi (1995) is a heuristic to solve the node-weighted variant of the Steiner tree problem. First, the distance between any node pair in the graph is obtained with an all-to-all shortest paths algorithm such as Dijkstra (1959). A set of trees is considered where each tree initially consists of a single seed node. At each step of the algorithm, a node and a subset of the remaining trees are selected such that the cost of tree merging is minimized. At least two trees have to be merged in each step. The cost of tree merging is computed as the sum of the weight of the selected node and the weights of the shortest paths between the selected node and the selected tree subset. This sum is divided by the number of trees in the selected subset. The algorithm terminates when all trees are merged. The same implementation as in Scott *et al.* (2005) has been used to evaluate this algorithm. The implementation was kindly provided by Nadja Betzler (Betzler (2005)). The computational complexity of this approach is $O(n^2 \log n + nm + ns^3 \log s)$.

2.4.4 Takahashi-Matsuyama The algorithm by Takahashi and Matsuyama (1980) initializes the sub-network with a node chosen at random among the s seeds. It then proceeds by identifying in each step the lightest path(s) between any of the remaining seed nodes and any node in the sub-network (note that pseudo nodes can be introduced to treat all nodes in the sub-network as equivalent start nodes and all remaining seed nodes as equivalent end nodes). The lightest path(s) is merged with the sub-network. The computational complexity of this approach is $O(s(m + Kn \log(m/n)))$.

2.4.5 Pair-wise K-shortest paths In the first step, REA is called successively on each pair of seed nodes. The resulting path sets are stored in a path matrix, and the minimal weight between each node pair is stored in a distance matrix. In the second step, the sub-network is constructed from the path sets, starting with the lightest path set. Step-wise, path sets are merged with the subgraph by increasing order of their weight. The process stops if either all seeds belong to one connected component of the sub-network or all path sets have been merged with the sub-network.

The computational complexity of this approach is $O(s^2(m + Kn \log(m/n))$, because the REA algorithm is called $O(s^2)$ times.

2.4.6 kWalks The kWalks method is a generic algorithm (Dupont *et al.* (2006)) to build a most relevant subgraph connecting seed nodes in a large graph, in the present case a metabolic network. The subgraph contains the most relevant edges and the nodes induced by those edges. The relevance of an edge is measured as the expected number of times it is visited along random walks connecting seed nodes. These expected passage times reflect both the topology of the network and the edge weights. They follow from an interpretation of the graph as a Markov chain (Kemeny and Snell (1983)) characterized by a transition probability matrix $\boldsymbol{P}$.

The probability of transition from node i to node j is given by $\boldsymbol{P}_{ij} = \frac{w_{ij}}{\sum_j w_{ij}}$ where w_{ij} denotes the weight assigned to the edge $i \rightarrow j$. For each seed node x, the sub-matrix ${}^x\boldsymbol{P}$ denotes the transition probability matrix restricted to the lines and columns associated to x and all non-seed nodes. Expected passage times can be computed from the fundamental matrix ${}^x\boldsymbol{N} = (\boldsymbol{I} - {}^x\boldsymbol{P})^{-1}$. The entry ${}^x\boldsymbol{N}_{xi}$

gives the expected number of times node i is visited during walks starting in x and ending in any other seed node. The expected passage times ${}^xE(i,j)$ along an edge $i \rightarrow j$ is obtained by multiplying ${}^x\boldsymbol{N}_{xi}$ with the transition probability $\boldsymbol{P}_{ij}$. Finally, the relevance of an edge $i \rightarrow j$ is obtained by averaging ${}^xE(i,j)$ over the s seed nodes.

A straightforward implementation of the kWalks algorithm is computationally demanding for a large graph: its complexity is $O(sn^3)$, since it would rely on s matrix inversions for a graph with n nodes. In practice, the fundamental matrix can however be approximated by limiting the walks to a maximal number of L steps and using forward-backward recurrences (Callut (2007)). The computational complexity of the bounded kWalks is $O(sLm)$. Since s, the number of seed nodes, as well as L are typically fixed and have values orders of magnitude lower than m, this approach essentially offers a linear time complexity with respect to the number of graph edges. Bounding the walk length is not only convenient from a computational viewpoint, it also allows to control the level of locality (or, conversely, the level of diffusion through the network) while connecting seed nodes. In all the reported experiments, L was fixed to 50 based on preliminary evaluations (Dupont *et al.* (2006)).

As such the kWalks algorithm computes edge and node relevance from random walks connecting the seed nodes. A subgraph is obtained by keeping only those edge above a minimal relevance threshold. In our experiments, the relevance threshold is automatically fixed such that the subgraph induced by the selected edges is weakly connected. The sub-networks extracted by kWalks may contain branches ending in non-seed nodes. We remove these branches in a final pruning step.

The edge relevances computed by kWalks can serve as new edge weights. kWalks can then be run on the input graph with updated weights. This iterative process may be repeated a number of times to increase the discrimination between more and less relevant edges.

2.4.7 Hybrid approaches On one hand, the kWalks approach is designed to be more sensitive than specific by returning a sub-network whose edges are more likely to be used along walks connecting the seed nodes. Such a sub-network may be significantly smaller than the initial network yet not highly specific to form relevant pathways. On the other hand, the computational complexity of path-based approaches may prevent them from being effective when applied to a large network. Those observations motivate the use of an hybrid strategy where the kWalks method is combined with paths-based algorithms. Such a hybrid approach runs in two steps: kWalks extracts a sub-network representing a fixed proportion of the input network and the shortest-path based algorithm is launched on this intermediate sub-network to obtain the final pathway.

Combining kWalks with path-based approaches requires two new parameters: (1) **Size of the sub-network** kWalks extracts a sub-network whose size is fixed to a given percentage of the number of nodes in the input network. In our experiments, this parameter is usually fixed between .5% and 5%. The extracted sub-networks tend to be larger than with the weak connectivity constraint but are subsequently filtered with a path-based approach. (2) **Input or computed weights** The path-based algorithms may either use the input weights or the edge/node relevances computed by kWalks.

2.5 Evaluation procedure

2.5.1 Accuracy of sub-network extraction Each pathway inference algorithm receives as seed nodes a varying number of reactions of the known pathway. The accuracy of the algorithm is then calculated based on the overlap between the extracted sub-network (that is the inferred pathway) and the reference pathway.

We define as true positive TP a non-seed node that is present in the reference as well as the inferred pathway. A false negative FN is a non-seed node present in the reference but missing in the inferred pathway and a false positive FP is a non-seed node found in the inferred pathway but absent from the reference. The sensitivity Sn is defined as the ratio of correctly inferred nodes versus all reference nodes:

$Sn = \frac{TP}{(TP+FN)}$, whereas the positive predictive value PPV gives the ratio of correctly inferred nodes versus all inferred nodes: $PPV = \frac{TP}{(TP+FP)}$. We calculate the accuracy as the geometric mean between sensitivity and positive predictive value ($Acc_g = \sqrt{Sn * PPV}$).

2.5.2 Experiments For each reference pathway, several inferences (i.e. sub-network extractions) are performed, with increasing seed node number, in order to test the impact of the seed node number on the accuracy of the result. For each of the 71 reference pathways, we first select the terminal reactions as seeds, we infer a pathway that interconnects them, and we compare the nodes of the inferred pathways with those of the annotated pathway. Then, we progressively increase the number of seeds by adding reactions randomly selected from the reference pathway, and re-do the inference and evaluation, until all reactions of the pathway are selected as seeds. We define as one experiment the set of all the pathway inferences performed for a given parameter value combination (e.g. pair-wise K-shortest paths on directed MetaCyc network with compound degree weight). In total, we carried out 108 such experiments.

3 RESULTS

3.1 Global performance of pathway inference algorithms

3.1.1 Comparison of algorithms The average geometric accuracy of a selected number of experiments is listed in Table 1. The full experiment table is available as supplementary Table ST1. The strategy resulting in the highest accuracy combines the Takahashi-Matsuyama algorithm with kWalks. The top experiments all involve a compound-weighted, directed MetaCyc network and, in case of the kWalks algorithm, an iteration number larger than one.

The performance of paths-based algorithms in the unweighted (unit weight), directed MetaCyc network is at most 53% whereas kWalks (without iteration) reaches an average accuracy of 62% in the same conditions. Hence kWalks is able to assign edge relevances even without a dedicated weight policy for the problem at hand, such as the compound degree weighting scheme for metabolic networks. All approaches however benefit from such a dedicated weight policy.

In the pair-wise K-shortest paths/kWalks hybrid approach, kWalks is configured to extract 5% of the input network. If this percentage is reduced to 0.5% (the optimum among 22 different sub-network sizes tested), the average accuracy increases by 3%. Obviously, the size of the intermediate sub-network should not go below a certain limit as it should be large enough to contain a metabolic pathway.

Combining a paths-based algorithm with kWalks tends to reduce its runtime. Supplementary Figure SF1 compares run-times for all 7 pathway inference algorithms.

3.1.2 Influence of parameter setting We measured the impact of alternative parameter values over a subset of the experiments as measured by a paired signed Wilcoxon rank test (Supplementary Table ST2). The parameter values having highest impact on the pathway inference accuracy are in this order: compound degree weight and inflated compound degree weight outperform unit weight, directed network outperforms undirected network, kWalks supersedes hybrid approaches and three kWalk iterations are better than a single run.

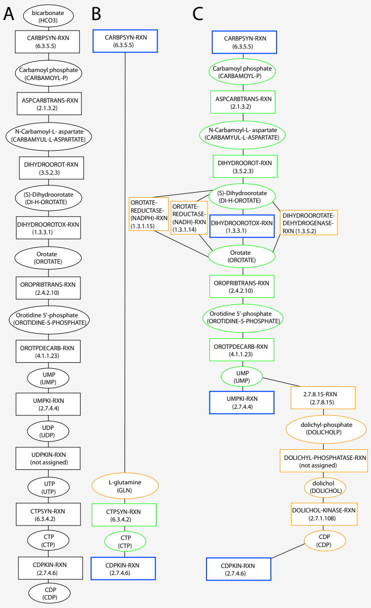

A
bicarbonate (HCO3)
CARBPSYN-RXN (6.3.5.5)
Carbamoyl phosphate (CARBAMOYL-P)
ASPCARBTRANS-RXN (2.1.3.2)
N-Carbamoyl-L- aspartate (CARBAMYUL-L-ASPARTATE)
DIHYDROOROT-RXN (3.5.2.3)
(S)-Dihydroorotate (DI-H-OROTATE)
DIHYDROOROTOX-RXN (1.3.3.1)
Orotate (OROTATE)
OROPRIBTRANS-RXN (2.4.2.10)
Orotidine 5'-phosphate (OROTIDINE-5-PHOSPHATE)
OROTPDECARB-RXN (4.1.1.23)
UMP (UMP)
UMPKI-RXN (2.7.4.4)
UDP (UDP)
UDPKIN-RXN (not assigned)
UTP (UTP)
CTPSYN-RXN (6.3.4.2)
CTP (CTP)
CDPKIN-RXN (2.7.4.6)
CDP (CDP)
B
CARBPSYN-RXN (6.3.5.5)
L-glutamine (GLN)
CTPSYN-RXN (6.3.4.2)
CTP (CTP)
CDPKIN-RXN (2.7.4.6)
C
CARBPSYN-RXN (6.3.5.5)
Carbamoyl phosphate (CARBAMOYL-P)
ASPCARBTRANS-RXN (2.1.3.2)
N-Carbamoyl-L- aspartate (CARBAMYUL-L-ASPARTATE)
DIHYDROOROT-RXN (3.5.2.3)
(S)-Dihydroorotate (DI-H-OROTATE)
OROTATE-REDUCTASE-(NADPH)-RXN (1.3.1.15)
OROTATE-REDUCTASE-(NADH)-RXN (1.3.1.14)
DIHYDROOROTOX-RXN (1.3.3.1)
DIHYDROOROTATE-DEHYDROGENASE-RXN (1.3.5.2)
Orotate (OROTATE)
OROPRIBTRANS-RXN (2.4.2.10)
Orotidine 5'-phosphate (OROTIDINE-5-PHOSPHATE)
OROTPDECARB-RXN (4.1.1.23)
UMP (UMP)
UMPKI-RXN (2.7.4.4)
2.7.8.15-RXN (2.7.8.15)
dolichyl-phosphate (DOLICHOLP)
DOLICHYL-PHOSPHATASE-RXN (not assigned)
dolichol (DOLICHOL)
DOLICHOL-KINASE-RXN (2.7.1.108)
CDP (CDP)
CDPKIN-RXN (2.7.4.6)

Fig. 1. Pathway inference results for the pyrimidine ribonucleotides de novo biosynthesis pathway (MetaCyc identifier: PWY0-162) in *E. coli*. (A) Reference pathway. (B) Pathway inferred with two seeds in the compound-weighted, directed MetaCyc network. (C) Pathway inferred with four seeds in the same network. Ellipses represent compounds, rectangles reactions. Compounds and reactions are labeled with their MetaCyc identifiers in capital letters, compounds in addition with their name and reactions with their associated EC number. Seed nodes have a blue border, true positive nodes a green and false positives an orange border.

The superiority of the other two weighting schemes over unit weights is in agreement with previous results (Croes *et al.* (2005, 2006)), which show that weighting the metabolic network avoids irrelevant hub compounds. It is also no surprise that the directed MetaCyc network yields higher accuracies than the undirected one, because the directed network prevents the traversal from substrate to substrate or from product to product.

It might seem surprizing that when all experiments are taken together, kWalks alone outperforms the pair-wise K-shortest paths hybrid, whereas the 5 top-raking approaches rely either on hybrid approach or path finding alone. The reason is that kWalks, as explained above, deals well with the unit weight policy, whereas the hybrid only performs well if it can either use weights generated by kWalks or by a weight policy that penalizes hub compounds. However, if run with optimal parameter values, both algorithms are among the top experiments (see Table 1). Iterating kWalks improves the accuracy, as it increases the difference between relevant and irrelevant edges.

3.2 Study cases

All study cases were inferred with the hybrid algorithm combining Takahashi-Matsuyama and kWalks in the directed, compound-weighted MetaCyc network.

Since we cannot infer reaction directions due to the way we constructed the MetaCyc network, inferred pathways are displayed

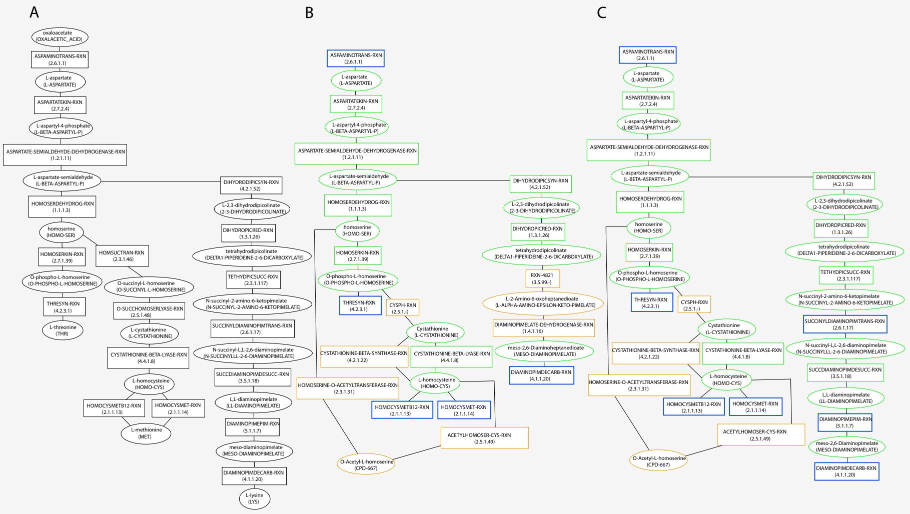


Fig. 2. Pathway inference results for the superpathway of lysine, threonine and methionine biosynthesis I (MetaCyc identifier: P4-PWY) in *E. coli*. (A) Reference pathway. (B) Pathway inferred with the five terminal reactions as seeds in the compound-weighted, directed MetaCyc network and Figure. (C) Pathway inferred with the terminal and two additional intermediate reactions in the same network. Ellipses represent compounds, rectangles reactions. Compounds and reactions are labeled with their MetaCyc identifiers in capital letters, compounds in addition with their name and reactions with their associated EC number. Seed nodes have a blue border, true positive nodes a green and false positives an orange border.

Table 1. Selected set of experiments, their conditions and results. Each table row represents one experiment. Each experiment was performed on 71 reference pathways with varying seed reaction number, comprising 406 launches of the tested pathway inference algorithm for the indicated conditions. Abbreviations: PPV = positive predictive value, acc.g = geometric accuracy

Algorithm	kWalks iteration number	Size of sub-network extracted by kWalks in %	Weighting scheme	Directed graph	kWalks weights re-used	Mean Sn in %	Mean PPV in %	Mean acc.g in %
Takahashi-Matusyama/kWalks	1	5	Compound degree	TRUE	FALSE	77.13	77.97	76.81
Takahashi-Matsuyama	0	-	Compound degree	TRUE	-	75.90	77.25	75.83
pair-wise K-shortest paths/kWalks	1	0.5	Compound degree	TRUE	FALSE	68.89	78.90	71.79
pair-wise K-shortest paths/kWalks	6	5	Compound degree	TRUE	FALSE	70.20	69.10	68.22
pair-wise K-shortest paths	0	-	Compound degree	TRUE	-	69.95	68.73	68.03
kWalks	3	-	Compound degree	TRUE	-	71.49	68.54	67.96
kWalks	6	-	Inflated compound degree	TRUE	-	71.06	68.62	67.90
pair-wise K-shortest paths/kWalks	3	5	Compound degree	TRUE	FALSE	69.19	69.37	67.86
Klein-Ravi/kWalks	1	5	Compound degree	FALSE	FALSE	63.21	68.03	64.10
kWalks	3	-	Unit	TRUE	-	61.40	71.33	64.30
kWalks	6	-	Unit	TRUE	-	60.00	71.75	63.53
Klein-Ravi	0	-	Compound degree	FALSE	-	62.55	66.27	63.05
kWalks	1	-	Unit	TRUE	-	62.13	65.93	61.83
pair-wise K-shortest paths/kWalks	1	5	Unit	TRUE	TRUE	46.91	69.38	55.32
Takahashi-Matsuyama	0	-	Unit	TRUE	-	60.02	53.83	52.74
pair-wise K-shortest paths	0	-	Unit	TRUE	-	71.37	35.87	42.86

as undirected graphs. The annotated pathways have been obtained from EcoCyc version 13.1 (Keseler *et al.* (2009)).

3.2.1 De novo synthesis of pyrimidine ribonucleotides in Escherichia coli The de novo synthesis of pyrimidine ribonucleotides pathway in *E. coli* produces CDP from L-glutamine in a series of 10 subsequent reaction steps (Figure 1A).

Two-end path finding results in a metabolic pathway that bypasses a large segment of the annotated pathway by taking a shortcut via L-glutamine (Figure 1B). Consequently, the geometric accuracy is low (28%).

With two additional seed nodes (Figure 1C), a large part of the reference pathway is recovered (geometric accuracy reaches 59%).

Not surprizingly, the result is more accurate when more information can be provided in the form of additional seed nodes. Such additional information could however add spurious paths between seed nodes, hence decreasing PPV, but the overall effect is clearly positive in this case.

3.2.2 Lysine, threonine and methionine biosynthesis in Escherichia coli The previous example illustrates the benefit of multi-seed pathway inference in the case of linear pathways. Another interest of the approach is its capacity to deal with branched metabolic pathways or super-pathways.

The lysine, threonine and methionine biosynthesis super-pathway of *E. coli* is a good example of a branched pathway that cannot be treated with two-end path finding (Figure 2A). This pathway starts from Oxaloacetate, the common precursor of the three amino acids L-lysine, L-methionine and L-threonine. The pathway is linear up to L-aspartyl-semialdehyde, after which it banches towards the three different end products. The synthesis of L-aspartyl-semialdehyde from L-aspartate is catalyzed by three isoenzymes (aspartate kinase I, II and III), each being negatively regulated by one of the three final products, thereby ensuring differential feedback inhibition. The annotated pathway consists of 18 reactions and 14 compounds, not counting the terminal compounds oxaloacetate, L-lysine, L-methionine and L-threonine.

Given the terminal reactions with MetaCyc identifiers
ASPAMINOTRANS-RXN, THRESYN-RXN,
DIAMINOPIMDECARB-RXN, HOMOCYSMETB12-RXN
and HOMOCYSMET-RXN, the pathway shown in Figure 2B is inferred from the MetaCyc network. It recovers large parts of the reference pathway, but misses parts of the annotated lysine and threonine branches, resulting in a geometric accuracy of 65%.

However, the inferred lysine branch is a biochemically valid metabolic pathway, which is known to be active e.g. in *Clostridium tetani* (MetaCyc pathway identifier: PWY-2942). Additional seed reactions are needed to distinguish the *E. coli* variant of lysine biosynthesis from this alternative. When repeating pathway inference with 2 additional reactions from the lysine branch (DIAMINOPIMEPIM-RXN and SUCCINYLDIAMINOPIMTRANS-RXN), the *E. coli* lysine biosynthesis pathway is found (see Figure 2C) and the geometric accuracy reaches 85%.

In practice, changes in the expression of regulated enzymes may reveal intermediate steps of a pathway. For instance, microarray experiments may reveal clusters of enzymes showing a transcriptional response to a given condition. Such expression clusters are likely to include terminal as well as intermediate enzymes, such as the gene argD associated to the intermediate seed reaction SUCCINYLDIAMINOPIMTRANS-RXN, which is negatively regulated by the transcription factor ArgR.

4 DISCUSSION

In this article, we presented different sub-network extraction techniques that can be applied to predict metabolic pathways from metabolic networks on the sole basis of network topology. The performance of these techniques was studied in metabolic networks, but they could be applied to any biological network.

From our evaluation we can conclude that a combination of Takahashi-Matsuyama and kWalks is globally most suited for the extraction of sub-networks from metabolic networks. The evaluation also shows that a directed, weighted metabolic network performs better than an undirected, unweighted one. Consequently, if a good weight policy for the metabolic network under study is at hand, it should be given as input to both algorithms, else the path-based algorithm can be launched on the weights computed by kWalks. The accuracy of pathway inference can be further increased by iterating kWalks and/or by reducing the size of the sub-network extracted by kWalks in the first step of the hybrid.

The hybrid approach combines the strengths of two different sub-network extraction strategies: kWalks is designed to capture the part of a network that is most relevant to connect the given seed nodes, resulting in a high sensitivity, but at the cost of a low positive predictive value. False positives introduced by kWalks can be discarded by a more stringent shortest paths-based algorithm.

Metabolic sub-network extraction can be applied to predict metabolic pathways for an organism whose genes are functionally annotated but whose metabolism is not yet known. In such a case, a network constructed from metabolic information taken from related organisms might be more appropriate than a complete metabolic network containing all known reactions and compounds in a given database as in this study. There are two ways to construct an organism-specific metabolic network: The first is to simply build the network from reactions occurring in the selected set of organisms. In a less restrictive approach, the complete network could be weighted in such a way that reactions occurring in the given organisms are favored over other reactions. Similarly, gene expression and other high-throughput data could be taken into account during network construction by converting expression ratios (or other scores derived from the data set) into node weights.

Pathway prediction could be further improved by taking into account the compound structure and atom flow through a reaction in order to distinguish main from side compounds. The RPAIR database available in KEGG provides the information required for this (Kotera *et al.* (2004a,b)). For two-node path finding, we already quantified the improvement due to RPAIRs (Faust *et al.* (2009)). First tests showed a similar improvement for sub-network extraction with multiple seeds in the RPAIR network as compared to the MetaCyc network.

Our pathway prediction approach is subjected to a number of limitations. Path-based approaches only partly infer cyclic or spiral-shaped pathways (the same enzymes acting repeatedly on a growing chain, e.g. fatty acids biosynthesis). kWalks alone is able to return general subgraphs but possibly at the cost of decreasing specificity. For certain pathways situated in the densely interconnected region of the metabolic network (such as the TCA cycle and the glycolysis pathway), a large number of seed nodes is required in order to distinguish them from alternative pathways. In addition, prediction accuracy is of course dependent on data quality. In order to infer a metabolic pathway from a metabolic network, the network must contain all nodes and edges of the pathway.

We rely on a topological definition that considers a metabolic pathway as a specific part of a metabolic network (e.g. Forst and

Schulten (1999)). This definition covers all classical pathways described in the literature, but also includes pathways that are not biochemically valid in contrast to stoichiometry-centered definitions (e.g. elementary modes, extreme pathways, see Schilling *et al.* (2000)). A revised definition of metabolic pathway could influence and possibly improve pathway prediction tools, but this objective is beyond the scope of the present article.

In a future application, we will apply the techniques evaluated in this article to gene expression data as well as operons, fusion genes and other data sets featuring genes assumed to be functionally related. This requires testing different scoring schemes in order to measure the quality of the predicted pathway.

The pathway inference algorithms were added to NeAT (Brohée *et al.* (2008)) at `http://rsat.ulb.ac.be/neat/`. A generic kWalks implementation is freely available at `www.ucl.ac.be/mlg/index.php?page=Softwares`.

ACKNOWLEDGEMENT

We would like to thank the former aMAZE team for their graph library and Nadja Betzler for providing us with her implementation of the Klein-Ravi algorithm. We acknowledge Jean-Noël Monette and Pierre Schaus from the INGI team at UCL for making publicly available their improvement of the REA code.

Funding: KF is supported by Actions de Recherches Concertées de la Communauté Française de Belgique (ARC grant number 04/09-307). The BiGRe laboratory is a member of the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265). The work was supported by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet).

REFERENCES

Antonov, A., Dietmann, S., and Mewes, H. W. (2008). KEGG spider: interpretation of genomics data in the context of the global gene metabolic network. *Genome Biology*, **9**.

Antonov, A., Dietmann, S., Wong, P., and Mewes, H. W. (2009). TICL - a web tool for network-based interpretation of compound lists inferred by high-throughput metabolomics. *FEBS Journal*, **276**, 2084–2094.

Arita, M. (2000). Metabolic reconstruction using shortest paths. *Simulation Practice and Theory*, **8**, 109–125.

Betzler, N. (2005). *Steiner Tree Problems in the Analysis of Biological Networks*. Master's thesis, Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany.

Blum, T. and Kohlbacher, O. (2008). MetaRoute: fast search for relevant metabolic routes for interactive network navigation and visualization. *Bioinformatics*, **24**, 2108–2109.

Brohée, S., Faust, K., Lima-Mendez, G., Sand, O., Janky, R., Vanderstocken, G., Deville, Y., and van Helden, J. (2008). NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Research*, **36**, W444–W451.

Callut, J. (2007). *First Passage Times Dynamics in Markov Models with Applications to HMM Induction, Sequence Classication, and Graph Mining*. Ph.D. thesis, Université catholique de Louvain.

Caspi, R., Foerster, H., Fulcher, C. A., Kaipa, P., Krummenacker, M., Latendresse, M., Paley, S., Rhee, S. Y., Shearer, A. G., Tissier, C., Walk, T. C., Zhang, P., and Karp, P. D. (2008). The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res*, **36**, D623–D631.

Croes, D., Couche, F., Wodak, S., and van Helden, J. (2005). Metabolic PathFinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Research*, **33**, W326–W330.

Croes, D., Couche, F., Wodak, S., and van Helden, J. (2006). Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.*, **356**, 222–236.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269–271.

Dittrich, M. T., Klau, G. W., Rosenwald, A., Dandekar, T., and Müller, T. (2008). Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, **24**, i223–i231.

Duin, C. W., Volgenant, A., and Voß, S. (2004). Solving group Steiner problems as Steiner problems. *European Journal of Operational Research*, **154**, 323–329.

Dupont, P., Callut, J., Dooms, G., Monette, J.-N., and Deville, Y. (2006). Relevant subgraph extraction from random walks in a graph. *Research Report UCL/FSA/INGI RR 2006-07*.

Ellis, L., Gao, J., Fenner, K., and Wackett, L. (2008). The University of Minnesota pathway prediction system: predicting metabolic logic. *Nucleic Acids Research*, **36**, W427–W432.

Faust, K., Croes, D., and van Helden, J. (2009). Metabolic path finding using RPAIR annotation. *Journal of Molecular Biology*, **388**, 390–414.

Forst, C. and Schulten, K. (1999). Evolution of metabolisms: A new method for the comparison of metabolic pathways using genomics information. *Journal of Computational Biology*, **6**, 343–360.

Hwang, F., Richards, D., and Winter, P. (1992). *The Steiner Tree Problem*, volume 53 of *Annals of Discrete Mathematics*. North-Holland, Amsterdam, Netherlands.

Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, **18**, S233–S240.

Jimenez, V. and Marzal, A. (1999). Computing the k shortest paths: a new algorithm and an experimental comparison. *Proc. 3rd Int. Worksh. Algorithm Engineering (WAE 1999)*, **1668**, 15–29.

Karp, R. (1972). *Reducibility among combinatorial problems*, pages 85–103. Complexity of Computer Computations. R. E. Miller and J. W. Thatcher, Plenum Press.

Kemeny, J. G. and Snell, J. L. (1983). Finite Markov Chains. *Springer-Verlag*.

Keseler, I. M., Bonavides-Martínez, C., Collado-Vides, J., Gama-Castro, S., Gunsalus, R. P., Johnson, D. A., Krummenacker, M., Nolan, L. M., Paley, S., Paulsen, I. T., Peralta-Gil, M., Santos-Zavaleta, A., Shearer, A. G., and Karp, P. D. (2009). EcoCyc: a comprehensive view of escherichia coli biology. *Nucleic Acids Res*, **37**, D464–D470.

Klein, P. and Ravi, R. (1995). A nearly best-possible approximation algorithm for node-weighted Steiner trees. *Journal of Algorithms*, **19**, 104–115.

Kotera, M., Okuno, Y., Hattori, M., Goto, S., and Kanehisa, M. (2004a). Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. *J Am Chem Soc.*, **126**, 16487–16498.

Kotera, M., Hattori, M., Oh, M.-A., Yamamoto, R., Komeno, T., Yabuzaki, J., Tonomura, K., Goto, S., and Kanehisa, M. (2004b). RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions. *Genome Informatics*, **15**, P062.

Krieger, C. J., Zhang, P., Mueller, L. A., Wang, A., Paley, S., Arnaud, M., Pick, J., Rhee, S. Y., and Palsson, B. Ø. (2004). MetaCyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res*, **32**, D438–D442.

McShan, D., Rao, S., and Shah, I. (2003). PathMiner: predicting metabolic pathways by heuristic search. *Bioinformatics*, **19**, 1692–1698.

Noirel, J., Ow, S. Y., Sanguinetti, G., Jaramillo, A., and Wright, P. C. (2008). Automated extraction of meaningful pathways from quantitative proteomics data. *Briefings in Functional Genomics and Proteomics*, **7**, 136–146.

Rahman, S., Advani, P., Schunk, R., Schrader, R., and Schomburg, D. (2004). Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). *Bioinformatics*, **21**, 1189–1193.

Rajagopalan, D. and Agarwal, P. (2005). Inferring pathways from gene lists using a literature-derived network of biological relationships. *Bioinformatics*, **21**(5), 788–793.

Schilling, C., Letscher, D., and Palsson, B. (2000). Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology*, **203**, 229–248.

Scott, M. S., Perkins, T., Bunnell, S., Pepin, F., Thomas, D. Y., and Hallett, M. (2005). Identifying regulatory subnetworks for a set of genes. *Mol Cell Proteomics*, **4**(5), 683–692.

Takahashi, H. and Matsuyama, A. (1980). An approximate solution for the Steiner problem in graphs. *Math. Japonica*, **24**, 573–577.

Zien, A., Küffner, R., Zimmer, R., and Lengauer, T. (2000). Analysis of gene expression data with pathway scores. In *Proceedings of the International Conference of Intelligent Systems Molecular Biology*, pages 407–417.

4 Application of pathway discovery to a gene expression data set from *S. cerevisiae*

The previous chapters (2 and 3) presented the development and evaluation of two-end and multiple-end pathway prediction approaches. In this chapter, multiple-end pathway prediction is applied to a microarray study conducted on *S. cerevisiae*.

4.1 Biological background

S. cerevisiae is capable to grow on a variety of nitrogenous compounds as sole nitrogen source, among others ammonium and urea. The nitrogen required in anabolic reactions is provided by glutamine and glutamate, the major nitrogen donors in yeast, which are either imported from the environment or derived from catabolic reactions of other nitrogenous compounds. Thus, yeast cells can survive on a single nitrogen source by metabolizing it directly and/or by degrading it to ammonium, glutamate or both. From ammonium and glutamate, glutamine can be synthesized, which can then serve together with glutamate as nitrogen donor for anabolic reactions.

4.2 Gene expression data set

In order to elucidate the response of yeast cells to the presence of alternative nitrogen sources, Godard et al. performed a gene expression study, which measured the effects of 21 different nitrogen sources on the gene expression in *S. cerevisiae* [63].

In the original article, the experiment was conducted and the resulting data processed in the following way:

1. Yeast cells were grown on each of the 21 compounds as sole nitrogen source and the expression of all 5,690 yeast genes was quantified.

2. Gene expression ratios were calculated for each of the 20 nitrogen sources as the log-ratio between gene expression given the nitrogen source and gene expression given urea as nitrogen source (considered as reference).

3. Genes with a P-value of differential expression below 1/5,690 were discarded to correct for multiple testing. The P-value was calculated with the SAM method [160].

4. A matrix of expression ratios was obtained consisting of 20 columns (for the 20 nitrogen sources) and 390 rows (all genes significantly differentially expressed for at least one nitrogen source and with expression ratios obtained for at least 13 nitrogen sources).

5. Hierarchical clustering of this matrix (see Figure 4.1) revealed two main groups, where group A contains good nitrogen sources (which allow generation times ∼2h) and group B bad nitrogen sources (which support only slow growth). The 20 sources and their groups are listed in Table 4.1, where group C refers to all nitrogen sources that neither clustered with group A nor with group B nitrogen sources.

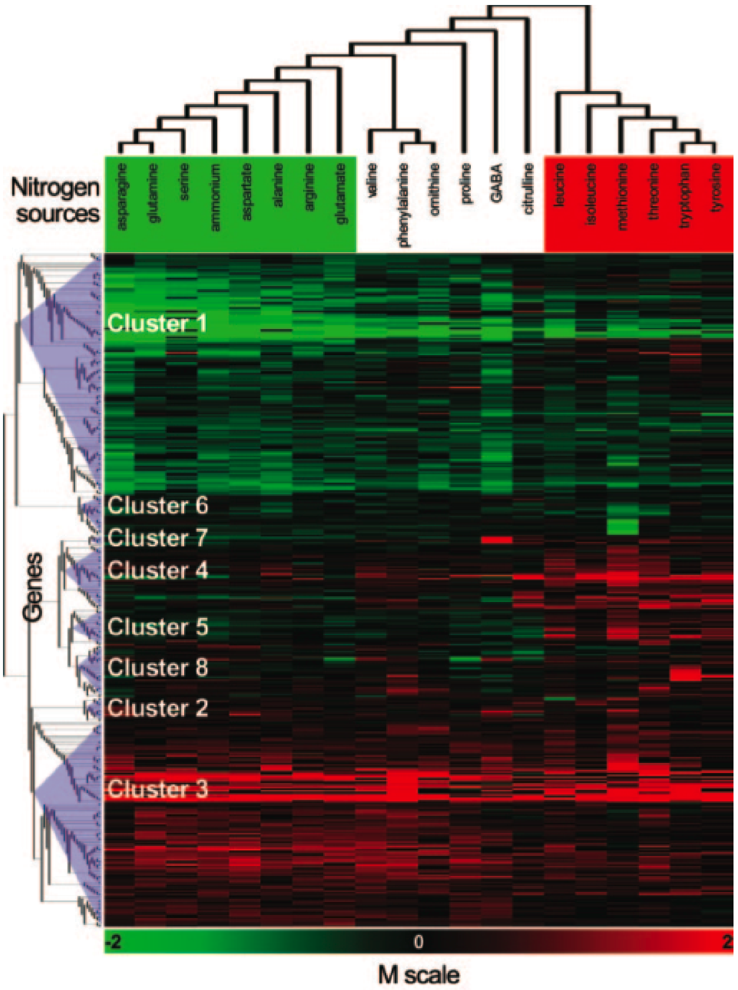


Figure 4.1: Gene expression ratio matrix re-arranged according to the outcome of the "complete linkage" cluster algorithm. The distances were calculated as the average dot product between gene expression profiles. Reproduction of Figure 2A from Godard et al., [63].

Table 4.1: Groups of nitrogen sources: A = good sources, B = bad sources, C = all other sources.

Nitrogen source (Abbreviation)	Group
asparagine (asn)	A
glutamine (gln)	A
serine (ser)	A
ammonium (amm)	A
aspartate (asp)	A
alanine (ala)	A
arginine (arg)	A
glutamate (glt)	A
valine (val)	C
phenylalanine (phe)	C
ornithine (orn)	C
proline (pro)	C
GABA (gab)	C
citrulline (cit)	C
leucine (leu)	B
isoleucine (ile)	B
methionine (met)	B
threonine (thr)	B
tryptophan (trp)	B
tyrosine (tyr)	B

The reference nitrogen source urea is stated in [63] to belong to the poor nitrogen sources.

4.3 Data processing

Given the results of Godard's experiments, I did the following analysis steps for each nitrogen source separately:

1. The distribution of gene expression ratios (i.e. the ratio between the gene expression value in the presence of the investigated nitrogen source with respect to urea) was plotted to check whether it approximates a Gauss distribution (see Figure 4.2). This was the case for all nitrogen sources.

2. Mean and standard deviation of the gene expression ratio distribution were robustly estimated with the median and interquartile range (IQR) using an R script written by Jacques van Helden.

3. The gene expression ratios were standardized as follows: $z_{i,j} = \frac{M_{i,j} - median(M_j)}{iqr(M_j)/k}$, where $M_{i,j}$ is the expression ratio of gene i for nitrogen source j; M_j denotes the vector of all expression ratios obtained for nitrogen source j (with 5,690 entries); k is a normalizing constant to estimate the standard deviation from the *iqr* [1]. The effect of the standardization is to convert a normal distribution $\mathcal{N}(\mu, \sigma)$ into a *standard normal* distribution $\mathcal{N}(0,1)$, i.e. a distribution centered on 0 and with a unit standard deviation. The standardized expression ratios are referred to as *z-scores*.

4. Each z-score $z_{i,j}$ is then converted into a *nominal P-value* by calculating the right tail of a normal distribution. The P-value $Pval_{i,j}$ is an estimation of the probability for a given gene (i) to reach by chance a given z-score ($z_{i,j}$) in a given microarray experiment (j). It is interpreted as the *risk of false positive*, i.e. the fact to erroneously consider as significant the expression ratio of a given gene i on a given microarray j.

5. For a large number of statistical tests (5,690 in this case), it is very likely that a test rejects the null hypothesis (i.e. no differential expression of a gene) by chance. For this reason, P-values are corrected for multiple testing using the Bonferroni correction, which converts P-values into E-values by multiplying each P-value with the number of tests performed, i.e. with the number of genes ($g = 5,690$):

$$Eval = Pval \cdot g$$

6. Finally, groups of up- and down-regulated genes are obtained by setting a threshold on the E-value. This is done separately for genes with positive z-scores (*up-regulated*) and negative z-scores (*down-regulated*), so that two gene groups are obtained for each nitrogen source. The E-value threshold was set to one, to restrict the number of genes falsely regarded as differentially expressed to one by nitrogen source. Since only a sub-set of all genes (namely the enzyme-coding genes) is considered for analysis, this threshold is sufficiently stringent to avoid false positives in most cases while preventing a loss of relevant enzymes.

[1] $k = qnorm(0.75) - qnorm(0.25) = 1.349$

After these processing steps, 20 up- and 20 down-regulated genes groups were obtained.

Table 4.2 summarizes the number of enzymes and their associated reactions and main reactant pairs for each gene group.

Table 4.2: Numbers of genes, enzymes and reactions associated to each gene group for an E-value threshold set to one

Gene cluster	Number of genes	Number of of enzymes	Number of EC numbers	Number of reactions	Number of main reactant pairs
ala_up	82	26	24	121	116
ala_down	84	24	25	88	71
amm_up	54	19	17	86	75
amm_down	66	21	22	88	78
arg_up	37	16	14	86	76
arg_down	66	19	21	53	56
asn_up	50	19	17	97	101
asn_down	95	33	34	106	92
asp_up	69	25	23	160	145
asp_down	48	12	14	28	24
cit_up	61	25	23	92	90
cit_down	27	6	6	31	25
gab_up	40	19	16	71	57
gab_down	71	19	22	79	58
gln_up	57	15	14	65	72
gln_down	69	26	28	80	78
glt_up	56	18	15	58	49
glt_down	64	24	25	109	94
ile_up	69	21	20	79	85
ile_down	23	6	6	34	24
leu_up	103	39	34	106	104
leu_down	26	9	8	17	16
met_up	119	49	45	179	200
met_down	33	9	10	41	33
orn_up	66	22	20	116	126
orn_down	39	14	14	57	52
phe_up	103	29	27	166	173
phe_down	24	4	4	10	13
pro_up	71	20	18	106	106
pro_down	32	9	11	18	17
ser_up	47	17	16	69	70
ser_down	59	17	18	65	73
thr_up	98	27	27	50	62
thr_down	16	2	2	2	3

Table 4.2: Numbers of genes, enzymes and reactions associated to each gene group for an E-value threshold set to one

Gene cluster	Number of genes	Number of of enzymes	Number of EC numbers	Number of reactions	Number of main reactant pairs
trp_up	121	47	38	136	138
trp_down	21	3	3	21	20
tyr_up	96	35	31	138	136
tyr_down	14	4	4	24	24
val_up	73	24	22	88	89
val_down	27	9	9	46	51

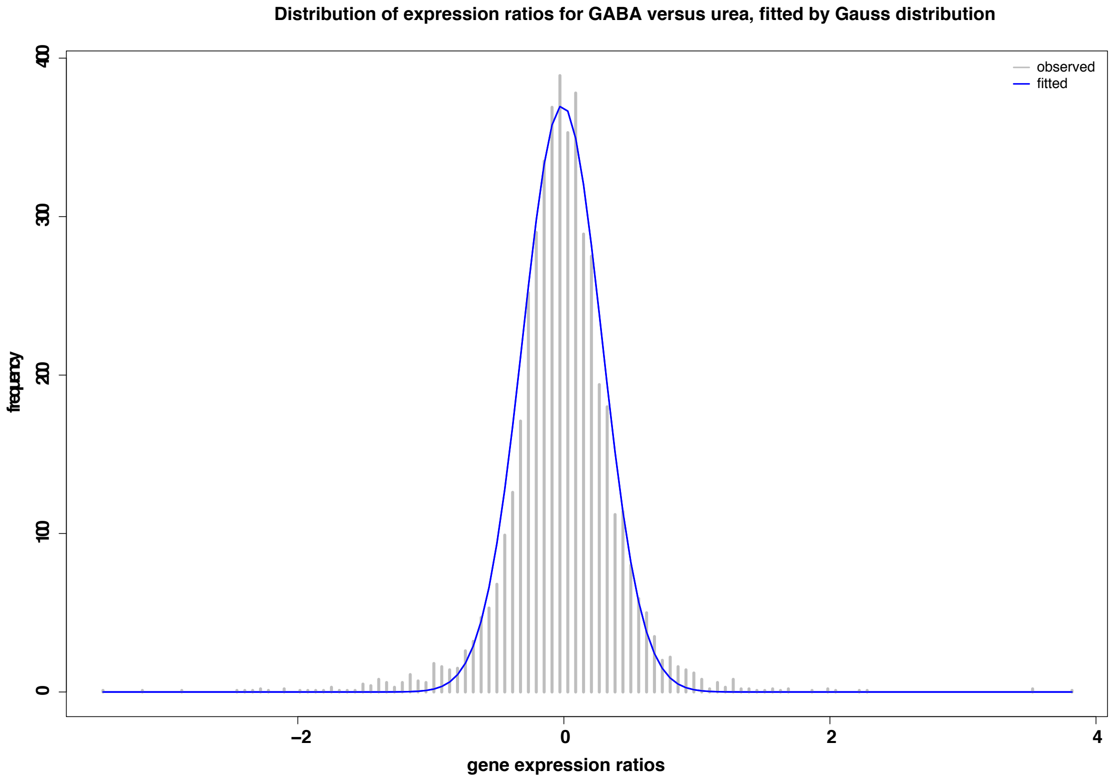


Figure 4.2: Distribution of gene expression ratios for GABA as sole nitrogen source with respect to urea. The blue line represents the Gauss distribution whose mean is estimated by the median of the gene expression ratios and whose standard deviation is estimated by their interquartile range. The estimated Gauss distribution describes well the gene expression ratio distribution. The Figure was generated with an R script written by Jacques van Helden.

4.4 Gene-to-reaction mapping

In order to predict a pathway from a group of genes, the genes have to be linked to their reactions first. As discussed in the Introduction, section 1.3.3, a many-to-many relationship exists between genes and reactions. Often, there is no direct link in the database from the enzyme-coding gene to the reactions its product catalyzes. In this case, reactions have to be obtained indirectly via the gene's EC number(s).

This complex relationship between genes and reactions poses several difficulties. First, an enzyme-coding gene may be connected to one or several EC numbers. In general, genes annotated with more than one EC number code for multifunctional enzymes, whose multiple catalytic sites are involved in the same pathway (e.g. the peroxisomal multifunctional enzyme type 2 in bile acid biosynthesis in rat). However, not all EC numbers associated to a gene are necessarily involved in the same pathway. For instance, the gene argD is associated to two EC numbers: 2.6.1.17 and 2.6.1.11. Of these two, only 2.6.1.17 contributes to the lysine biosynthesis pathway (MetaCyc identifier: DAPLYSINESYN-PWY), whereas 2.6.1.11 (but not 2.6.1.17) plays a role in the arginine biosynthesis pathway (MetaCyc identifier: ARGSYN-PWY).

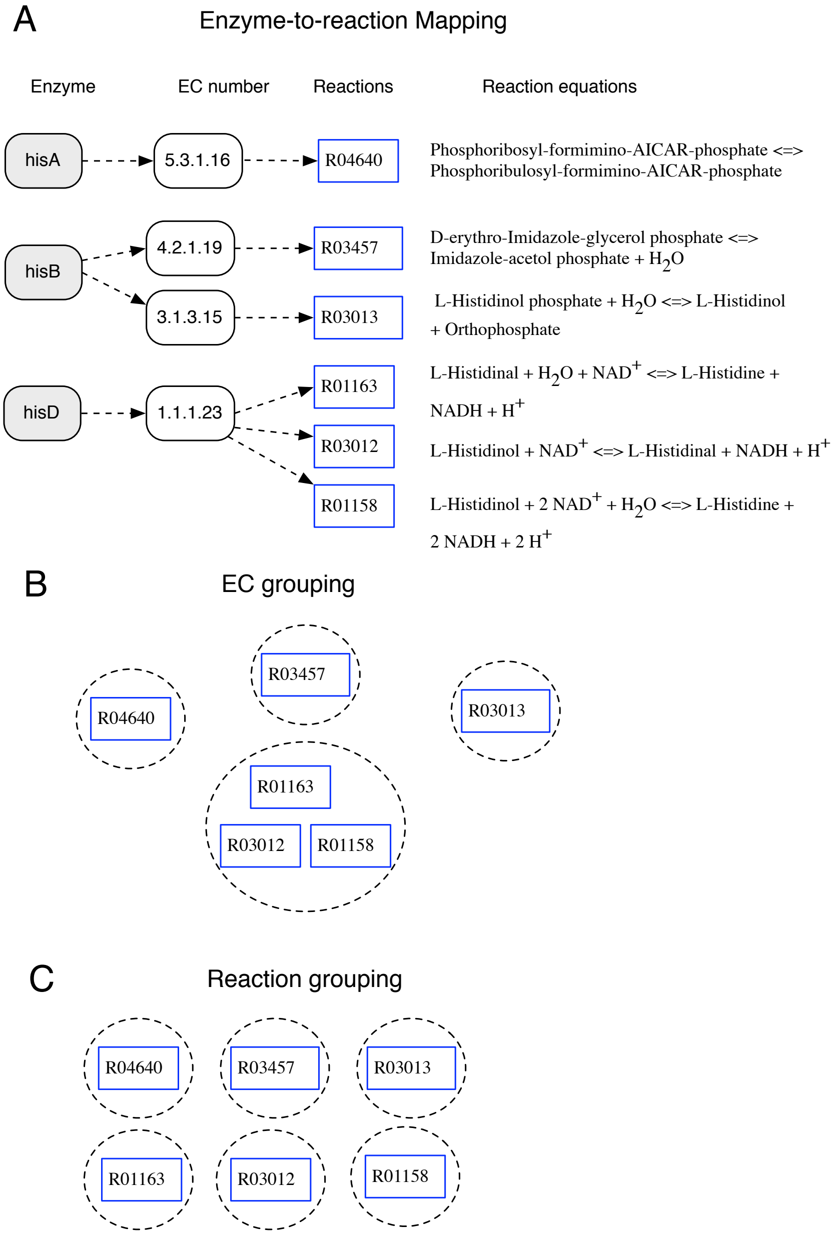
Second, each EC number is linked to one or more reactions. For example, EC number 1.1.1.1 (conversion of an alcohol into an aldehyde or ketone) is associated to 18 reactions in KEGG, out of which only one may be relevant for the pathway to be predicted. However, selecting only one out of a group of reactions associated to an EC number is not always a good strategy, because several EC numbers contribute more than one reaction to a pathway (this is the case for seven out of 55 reference pathways annotated in aMAZE for *E. coli*).

It is therefore an open question how reactions associated to genes should be grouped. Seed reactions can be grouped on the level of the genes, EC numbers or reactions, but as discussed, none of these groupings is correct in all cases. To evaluate which of these grouping strategies is correct in most cases, I performed a comparative evaluation on the 55 aMAZE *E. coli* pathways for two of them. The first strategy, called *reaction grouping*, treats all the reactions obtained from the enzyme-coding genes of a reference pathway as separate groups, whereas the second strategy, named *EC grouping*, treats each of the EC numbers of a reference pathway as a separate group, thus introducing an AND relationship between reactions in different EC number groups and an OR relationship between reactions belonging to the same EC number. Figure 4.3 illustrates the two grouping strategies.

In the first step of the evaluation, annotated genes of a reference pathway were mapped to their corresponding reactions. The reactions were then grouped into seed groups according to the selected grouping strategy. Next, the pathway was predicted from the seed reaction groups. Finally, the accuracy of the predicted pathway was computed. These steps were carried out for 55 pathways. Figure 4.4 illustrates this evaluation procedure.

Table 4.3 summarizes the results of this evaluation. The average accuracies are much lower than those obtained for two-end path finding evaluation (see chapter 2) for three reasons:

- Pathways were not linearized.
- Terminal compounds were not removed.



A
Enzyme-to-reaction Mapping
Enzyme
EC number
Reactions
Reaction equations
hisA
5.3.1.16
R04640
Phosphoribosyl-formimino-AICAR-phosphate <=> Phosphoribulosyl-formimino-AICAR-phosphate
hisB
4.2.1.19
R03457
D-erythro-Imidazole-glycerol phosphate <=> Imidazole-acetol phosphate + H_2O
3.1.3.15
R03013
L-Histidinol phosphate + H_2O <=> L-Histidinol + Orthophosphate
hisD
1.1.1.23
R01163
L-Histidinal + H_2O + NAD^+ <=> L-Histidine + NADH + H^+
R03012
L-Histidinol + NAD^+ <=> L-Histidinal + NADH + H^+
R01158
L-Histidinol + 2 NAD^+ + H_2O <=> L-Histidine + 2 NADH + 2 H^+
B
EC grouping
R04640
R03457
R03013
R01163
R03012
R01158
C
Reaction grouping
R04640
R03457
R03013
R01163
R03012
R01158

Figure 4.3: As an example, three selected enzymes from the histidine biosynthesis pathway are linked to their respective EC numbers and reactions (A). The EC grouping strategy groups reactions according to their EC numbers. In this example, four seed groups result, corresponding to the four EC numbers associated to the three enzymes (B). The reaction grouping strategy instantiates one group for each reaction. Thus, six seed groups result (C).

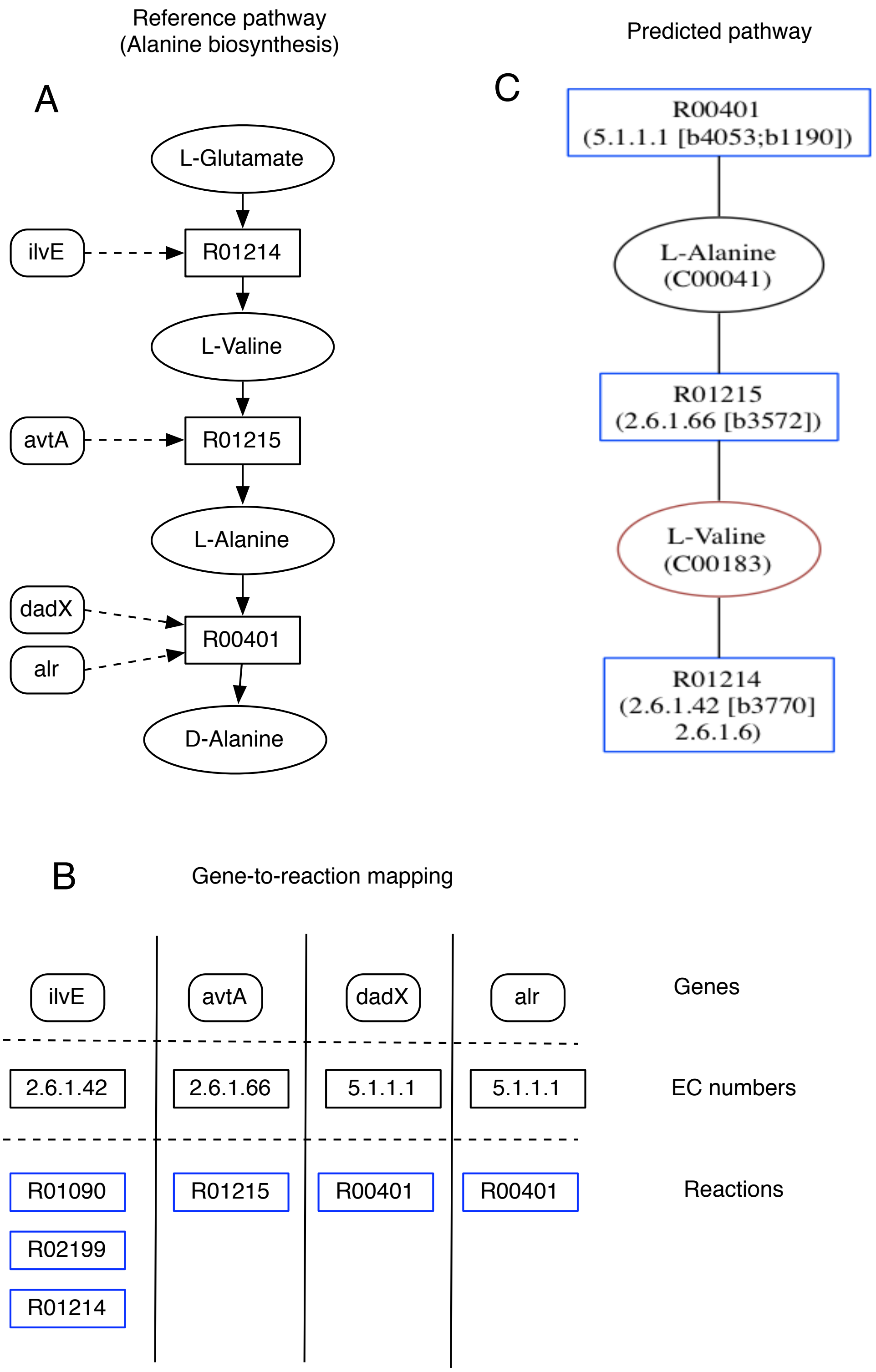


Figure 4.4: Procedure for the evaluation of reaction grouping strategies. The evaluation starts with a reference pathway (A). The genes of the reference pathway are mapped to reactions (B), which are grouped according to the strategy to be evaluated. In the example shown, reactions are grouped EC number-wise. From the seed reaction groups, a pathway is predicted (C), which is then compared to the reference pathway.

Table 4.3: Evaluation of seed reaction grouping strategies on 55 aMAZE pathways from *E. coli*

Network	Grouping strategy	Average geometric accuracy in %
KEGG LIGAND	Reaction groups	45
KEGG LIGAND	EC groups	44
KEGG RPAIR	Reaction groups	47
KEGG RPAIR	EC groups	50

Table 4.4: P-values of the paired signed Wilcoxon rank test for the two networks and grouping strategies. The number of pathways predicted with different accuracies is given in brackets.

	LIGAND EC groups	LIGAND reaction groups	RPAIR EC groups	RPAIR reaction groups
LIGAND EC groups	-	0.4 (35)	0.079 (32)	0.63 (39)
LIGAND reaction groups	0.4 (35)	-	0.19 (35)	0.47(36)
RPAIR EC groups	0.079 (32)	0.19 (35)	-	0.39 (40)
RPAIR reaction groups	0.63 (39)	0.47 (36)	0.39 (40)	-

- Gene-to-reaction mapping sometimes yields reactions that do not occur in the pathway (e.g. in Figure 4.4 reactions R01090 and R02199) and which may introduce false positive branches.

Table 4.4 lists the results of a paired signed Wilcoxon rank test, which was performed in order to check whether the accuracy difference between two networks or mapping strategies is significant.

From the evaluation, it can be concluded that:

- There is no significant difference between EC groups and reaction groups for the KEGG LIGAND network.

- There is no significant difference between EC groups and reaction groups for the KEGG RPAIR network.

- There is no significant difference between the KEGG LIGAND and RPAIR networks for the reaction grouping strategy, but for the EC grouping strategy, KEGG RPAIR performs significantly better than the KEGG LIGAND network.

Since the average geometric accuracy was highest for EC groupings in the KEGG RPAIR network, this grouping strategy and network was selected for pathway prediction.

4.5 Multiple-end pathway prediction parameters

4.5.1 Metabolic network

From the evaluation of gene-to-reaction mapping strategies, it emerged that the EC grouping in the KEGG RPAIR network performed best. Thus, the KEGG RPAIR network (KEGG RPAIR vs 49.0) was selected as input network for pathway prediction. It consists of 11,066 reactant pairs and 5,760 compounds, connected by 44,236 edges.

The MetaCyc network, which offers a more precise gene-to-reaction mapping (because genes can be directly linked to reactions), could not be employed, because MetaCyc does not contain many yeast genes.

4.5.2 Seed nodes

Two pathway predictions were performed for each nitrogen source: one for the up-regulated genes and the second for the down-regulated genes.

The gene groups were filtered such that for each group only the 5 enzyme-coding genes with the lowest E-value were retained. Previous experiments with unfiltered gene groups gave complex results that were hard to interpret. The aim of the reduction of the input gene number is to keep only the part of the pathway that is most affected by the nitrogen source in question.

For each of the reduced gene groups, reactant pairs were obtained in three steps using KEGG data:

1. For each enzyme-coding gene, EC numbers were obtained by querying the KEGG database on-the-fly.

2. Each EC number was associated to its reactions using the custom metabolic database described in section 9.3.

3. Each reaction in turn was associated to its main reactant pairs using the custom metabolic database.

4. The previous steps yield for each gene cluster a set of EC number groups, where each EC number group consists of a set of main reactant pairs. EC number groups with mutually exclusive reactant pairs (i.e. reactant pairs belonging to the same reaction) were merged into one group, to prevent that the same reaction appears twice in the predicted pathway.

5. Finally, overlapping groups (i.e. groups containing the same reactant pair) were merged as well.

4.5.3 Algorithm

Pathways were predicted with the hybrid of kWalks and Takahashi-Matsuyama, which was the best-performing algorithm in the evaluation presented in chapter 3. The intermediate network size extracted by kWalks in the first step of the hybrid was set to 5% of the input network size (i.e. the input network node number). KWalks was not iterated and its relevances not used as weights. Preprocessing was enabled (see section 9.1 for details on these parameters). Compounds adjacent to seed reactant pairs were included in the predictions.

After execution of the algorithm on the 40 gene clusters, 39 pathways could be predicted (for thr_down, which is associated to only three main reactant pairs, pathway prediction failed).

4.6 Pathway predicted to be up-regulated in the presence of aspartate

4.6.1 Up-regulated genes

In the presence of aspartate, 69 genes are up-regulated, 25 of them enzyme-coding. Table 4.5 displays the EC numbers and reactions associated to the five top enzyme-coding genes (i.e. those with lowest E-values). One of the genes, ADH4, codes for a broad-specificity enzyme that is associated to no less than 17 main reactant pairs. Seed reaction grouping allows to treat cases like this one, where only a subset of the reactions associated to a gene is likely to be relevant for the pathway.

Table 4.5: Gene-to-reactant pair mappings for the top five enzyme-coding genes up-regulated with aspartate as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
YCR012W	PGK1	3-phosphoglycerate kinase, catalyzes transfer of high-energy phosphoryl groups from the acyl phosphate of 1,3-bisphosphoglycerate to ADP to produce ATP	2.7.2.3	R01512 [RP00003, RP00113]
YER062C	HOR2	One of two redundant DL-glycerol-3-phosphatases (RHR2/GPP1 encodes the other) involved in glycerol biosynthesis	3.1.3.21	R00841 [RP00194] R07298 [RP01251]
YHR044C	DOG1	2-deoxyglucose-6-phosphate phosphatase, similar to Dog2p, member of a family of low molecular weight phosphatases	3.1.3.68	R08548 [RP00680] R02587 [RP02325]
YBR067C	TIP1	Major cell wall mannoprotein with possible lipase activity	3.1.1.-	R07680 [RP11273] R07677 [RP11937] R06729 [RP09164, RP09420] R05420 [RP05028]
YGL256W	ADH4	Alcohol dehydrogenase isoenzyme type IV, dimeric enzyme demonstrated to be zinc-dependent despite sequence similarity to iron-activated alcohol dehydrogenases	1.1.1.1	R05234 [RP11679] R00623 [RP00139] R07327 [] R05233 [RP04835] R08558 [RP00236] R00624 [RP00542] R06917 [RP09622] R07326 [] R08306 [RP13225] R08310 [RP13235] R06927 [RP09285]

Table 4.5: Gene-to-reactant pair mappings for the top five enzyme-coding genes up-regulated with aspartate as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
				R08557 [RP00236] R01041 [RP00374] R08281 [RP13149] R04880 [RP04492] R02124 [RP01983] R00754 [RP00238] R07105 [RP10191] R04805 [RP04436]

4.6.2 Predicted pathway

The pathway predicted from the five enzyme-coding genes most significantly up-regulated in the presence of aspartate as sole nitrogen source is shown in Figure 4.5. Overall, the pathway is not well supported by *S. cerevisiae* enzymes. Especially the branch leading to L-Galactono-1,4-lactone might be a false positive. A part of the pathway covers glycerol biosynthesis from D-glycerate, which could indicate up-regulated lipid biosynthesis.

Because of the many reactions not supported by *S. cerevisiae* genes, the prediction was repeated in a *S. cerevisiae* specific KEGG RPAIR network extracted from KEGG PATHWAY version 46. The result is depicted in Figure 4.6. The pathway consists of three components. Pathways extracted from KEGG RPAIR networks are sometimes disconnected due to mutual exclusion between reactant pairs (see chapter 2). The branch ending in L-Galactono-1,4-lactone present in the pathway predicted from the generic KEGG RPAIR network indeed disappears as well as the connection between 3-Phospho-D-glycerate and glycerol.

In summary, in the presence of aspartate, glycerol biosynthesis is predicted to be up-regulated. This might indicate that in the presence of the good nitrogen source aspartate, storage compound synthesis is up-regulated.

4.7 Pathway predicted to be down-regulated in the presence of aspartate

4.7.1 Down-regulated genes

In the presence of aspartate as sole nitrogen source, 12 out of 48 significantly down-regulated genes could be associated to EC numbers and reactions. Several of the down-regulated genes code for proteins of unknown function (e.g. YDR090C, YPL054W, YHR029C, YIR030C), which illustrates the fact that even for well studied organisms like *S. cerevisiae*, the proteome

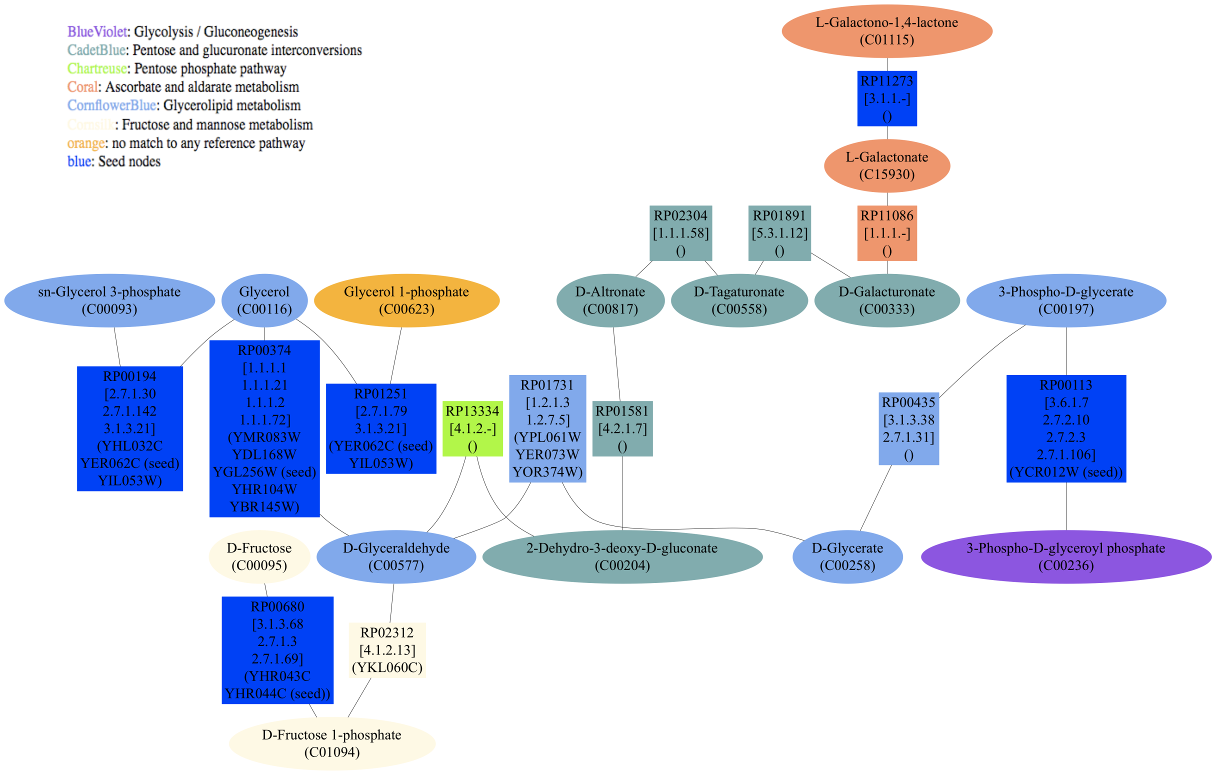

BlueViolet: Glycolysis / Gluconeogenesis
CadetBlue: Pentose and glucuronate interconversions
Chartreuse: Pentose phosphate pathway
Coral: Ascorbate and aldarate metabolism
CornflowerBlue: Glycerolipid metabolism
Cornsilk: Fructose and mannose metabolism
orange: no match to any reference pathway
blue: Seed nodes
L-Galactono-1,4-lactone (C01115)
RP11273 [3.1.1.-] ()
L-Galactonate (C15930)
RP02304 [1.1.1.58] ()
RP01891 [5.3.1.12] ()
RP11086 [1.1.1.-] ()
sn-Glycerol 3-phosphate (C00093)
Glycerol (C00116)
Glycerol 1-phosphate (C00623)
D-Altronate (C00817)
D-Tagaturonate (C00558)
D-Galacturonate (C00333)
3-Phospho-D-glycerate (C00197)
RP00194 [2.7.1.30 2.7.1.142 3.1.3.21] (YHL032C YER062C (seed) YIL053W)
RP00374 [1.1.1.1 1.1.1.21 1.1.1.2 1.1.1.72] (YMR083W YDL168W YGL256W (seed) YHR104W YBR145W)
RP01251 [2.7.1.79 3.1.3.21] (YER062C (seed) YIL053W)
RP13334 [4.1.2.-] ()
RP01731 [1.2.1.3 1.2.7.5] (YPL061W YER073W YOR374W)
RP01581 [4.2.1.7] ()
RP00435 [3.1.3.38 2.7.1.31] ()
RP00113 [3.6.1.7 2.7.2.10 2.7.2.3 2.7.1.106] (YCR012W (seed))
D-Fructose (C00095)
D-Glyceraldehyde (C00577)
2-Dehydro-3-deoxy-D-gluconate (C00204)
D-Glycerate (C00258)
3-Phospho-D-glyceroyl phosphate (C00236)
RP00680 [3.1.3.68 2.7.1.3 2.7.1.69] (YHR043C YHR044C (seed))
RP02312 [4.1.2.13] (YKL060C)
D-Fructose 1-phosphate (C01094)

Figure 4.5: The pathway predicted from the top five enzyme-coding genes up-regulated in the presence of aspartate as sole nitrogen source. Legend: Square=reactant pair (labeled with its KEGG identifier and associated EC numbers and genes), ellipse=compound (labeled with its KEGG identifier and name).

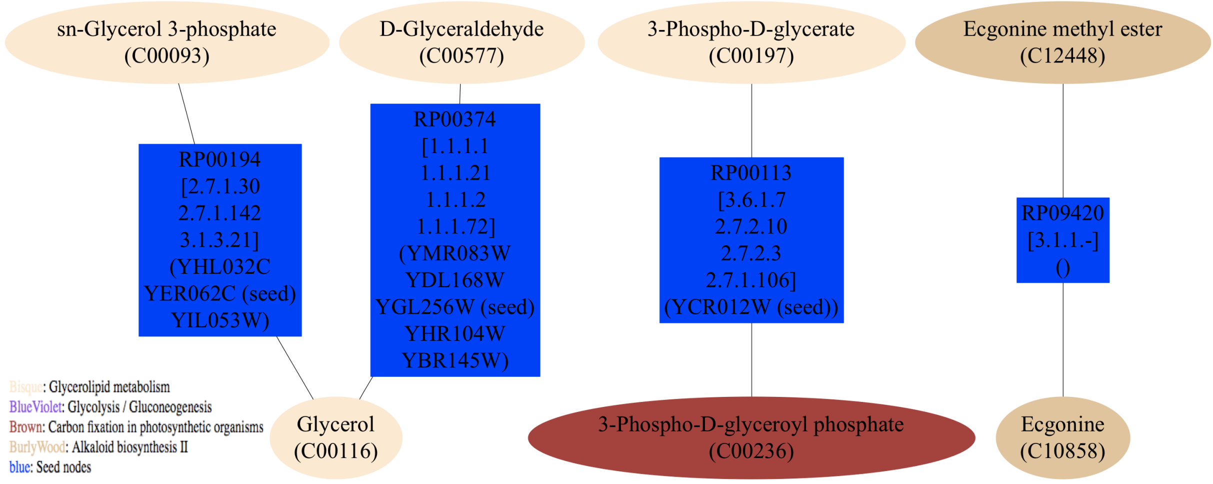

sn-Glycerol 3-phosphate (C00093)
D-Glyceraldehyde (C00577)
3-Phospho-D-glycerate (C00197)
Ecgonine methyl ester (C12448)
RP00194 [2.7.1.30 2.7.1.142 3.1.3.21] (YHL032C YER062C (seed) YIL053W)
RP00374 [1.1.1.1 1.1.1.21 1.1.1.2 1.1.1.72] (YMR083W YDL168W YGL256W (seed) YHR104W YBR145W)
RP00113 [3.6.1.7 2.7.2.10 2.7.2.3 2.7.1.106] (YCR012W (seed))
RP09420 [3.1.1.-] ()
Bisque: Glycerolipid metabolism
BlueViolet: Glycolysis / Gluconeogenesis
Brown: Carbon fixation in photosynthetic organisms
BurlyWood: Alkaloid biosynthesis II
blue: Seed nodes
Glycerol (C00116)
3-Phospho-D-glyceroyl phosphate (C00236)
Ecgonine (C10858)

Figure 4.6: The pathway predicted from the top five enzyme-coding genes up-regulated in the presence of aspartate as sole nitrogen source in a *S. cerevisiae* specific RPAIR network. Legend: Square=reactant pair (labeled with its KEGG identifier and associated EC numbers and genes), ellipse=compound (labeled with its KEGG identifier and name).

is not completely elucidated. Table 4.6 lists the associated EC numbers and reactions of the top five enzyme-coding genes.

Another annotation problem is apparent for gene YNL141W (AAH1): In KEGG, it is annotated with EC number 3.5.4.2 (gene definition field), but in fact is associated to EC number 3.5.4.4 according to both KEGG and SGD [74].

YCL064C (CHA1) is known to accept both serine (EC number: 4.3.1.17) and threonine (EC number: 4.3.1.19) as substrates and thus furnishes an example for a gene associated to more than one EC number.

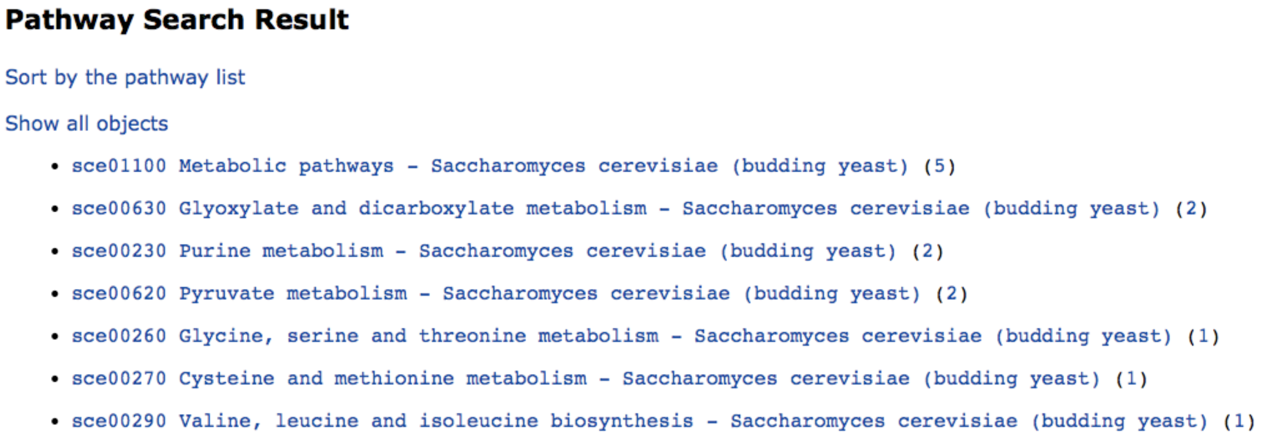
There are also two genes associated to the same EC number, namely YNL117W (MLS1) and YIR031C (DAL7). Both code for a malate synthase and are thus an example for isoenzymes.

Table 4.6: Gene-to-reactant pair mappings for for the top five enzyme-coding genes down-regulated with aspartate as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
YIR029W	DAL2	Allantoicase, converts allantoate to urea and ureidoglycolate in the second step of allantoin degradation	3.5.3.4	R02422 [RP02197, RP02198]
YCL064C	CHA1	Catabolic L-serine (L-threonine) deaminase, catalyzes the degradation of both L-serine and L-threonine	4.3.1.19	R00220 [RP04290] R00996 [RP01226] R06131 [RP00068]
			4.3.1.17	R00220 [RP04290] R00590 [RP00998] R06131 [RP00068]
YNL141W	AAH1	Adenine deaminase (adenine aminohydrolase), converts adenine to hypoxanthine	3.5.4.4	R01560 [RP01594] R06137 [] R02556 [RP02305]
YNL117W	MLS1	Malate synthase, enzyme of the glyoxylate cycle, involved in utilization of non-fermentable carbon sources	2.3.3.9	R00472 [RP00921, RP00007]
YIR031C	DAL7	Malate synthase, role in allantoin degradation unknown	2.3.3.9	R00472 [RP00921, RP00007]

4.7.2 Predicted pathway

The reactions associated to the 5 enzyme-coding genes are contained in 6 KEGG maps (see Figure 4.7), which underlines the difficulty of interpreting a differentially expressed gene group by pathway mapping alone.



Pathway Search Result

Sort by the pathway list

Show all objects

- sce01100 Metabolic pathways - Saccharomyces cerevisiae (budding yeast) (5)
- sce00630 Glyoxylate and dicarboxylate metabolism - Saccharomyces cerevisiae (budding yeast) (2)
- sce00230 Purine metabolism - Saccharomyces cerevisiae (budding yeast) (2)
- sce00620 Pyruvate metabolism - Saccharomyces cerevisiae (budding yeast) (2)
- sce00260 Glycine, serine and threonine metabolism - Saccharomyces cerevisiae (budding yeast) (1)
- sce00270 Cysteine and methionine metabolism - Saccharomyces cerevisiae (budding yeast) (1)
- sce00290 Valine, leucine and isoleucine biosynthesis - Saccharomyces cerevisiae (budding yeast) (1)

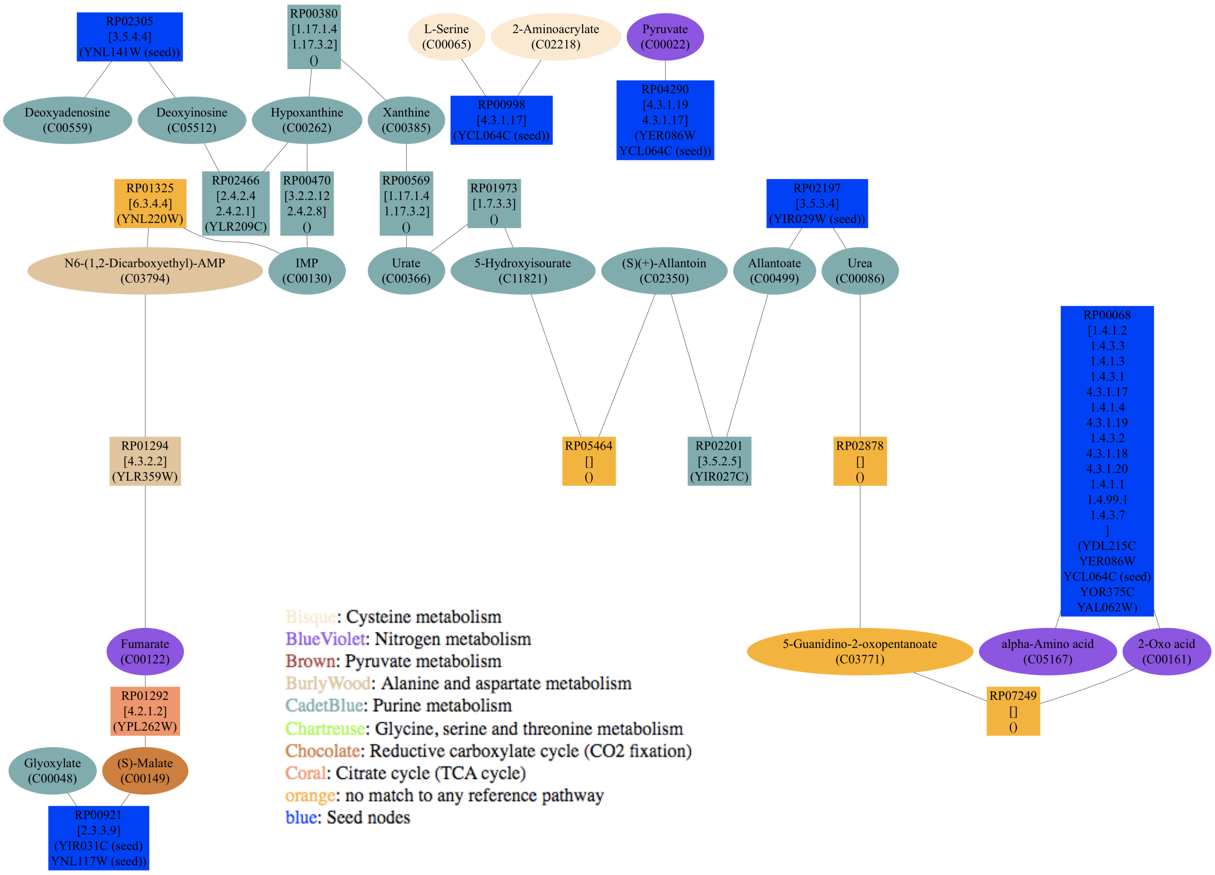
Figure 4.7: The list of pathways obtained from KEGG for the top five enzyme-coding genes down-regulated in the presence of aspartate as sole nitrogen source. *Metabolic pathways* is a global map, comprising all other KEGG maps.

The predicted pathway, shown in Figure 4.8, consists of three components: two of them are orphan seed reactant pairs with their adjacent compounds (RP00998 and RP04290).

The prediction suggests a degradation pathway for purines, which converts deoxyadenosine to deoxyinosine, hypoxanthine and xanthine, which is finally degraded to urate. This purine degradation pathway corresponds well to the known purine degradation pathway I in MetaCyc (identifier: PWY-5044). The purine degradation pathway is connected via 5-hydroxyisourate to the allantoin degradation pathway (MetaCyc identifier: PWY-5697), which yields urea. In addition, there are dubious connections to steps of the TCA cycle (fumarate, malate) and to a reaction converting the generic compound 2-oxo acid into the generic compound alpha-amino acid.

The prediction suggests that in the presence of the good nitrogen source aspartate, the degradation of purines and allantoin is suppressed.

The nitrogen catabolite repression (NCR) is known to suppress a number of nitrogen degradation pathways in the presence of good nitrogen sources in *S. cerevisiae* and other fungi (e.g. [170]). One may hypothesize that the aforementioned degradation pathways are also under control of the NCR.



RP02305 [3.5.4.4] (YNL141W (seed))
RP00380 [1.17.1.4 1.17.3.2] ()
L-Serine (C00065)
2-Aminoacrylate (C02218)
Pyruvate (C00022)
Deoxyadenosine (C00559)
Deoxyinosine (C05512)
Hypoxanthine (C00262)
Xanthine (C00385)
RP00998 [4.3.1.17] (YCL064C (seed))
RP04290 [4.3.1.19 4.3.1.17] (YER086W YCL064C (seed))
RP01325 [6.3.4.4] (YNL220W)
RP02466 [2.4.2.4 2.4.2.1] (YLR209C)
RP00470 [3.2.2.12 2.4.2.8] ()
RP00569 [1.17.1.4 1.17.3.2] ()
RP01973 [1.7.3.3] ()
RP02197 [3.5.3.4] (YIR029W (seed))
N6-(1,2-Dicarboxyethyl)-AMP (C03794)
IMP (C00130)
Urate (C00366)
5-Hydroxyisourate (C11821)
(S)(+)-Allantoin (C02350)
Allantoate (C00499)
Urea (C00086)
RP00068 [1.4.1.2 1.4.3.3 1.4.1.3 1.4.3.1 4.3.1.17 1.4.1.4 4.3.1.19 1.4.3.2 4.3.1.18 4.3.1.20 1.4.1.1 1.4.99.1 1.4.3.7] (YDL215C YER086W YCL064C (seed) YOR375C YAL062W)
RP01294 [4.3.2.2] (YLR359W)
RP05464 [] ()
RP02201 [3.5.2.5] (YIR027C)
RP02878 [] ()
Fumarate (C00122)
5-Guanidino-2-oxopentanoate (C03771)
alpha-Amino acid (C05167)
2-Oxo acid (C00161)
RP01292 [4.2.1.2] (YPL262W)
RP07249 [] ()
Glyoxylate (C00048)
(S)-Malate (C00149)
RP00921 [2.3.3.9] (YIR031C (seed) YNL117W (seed))
Bisque: Cysteine metabolism
BlueViolet: Nitrogen metabolism
Brown: Pyruvate metabolism
BurlyWood: Alanine and aspartate metabolism
CadetBlue: Purine metabolism
Chartreuse: Glycine, serine and threonine metabolism
Chocolate: Reductive carboxylate cycle (CO2 fixation)
Coral: Citrate cycle (TCA cycle)
orange: no match to any reference pathway
blue: Seed nodes

Figure 4.8: The pathway predicted from the top five enzyme-coding genes down-regulated in the presence of aspartate as sole nitrogen source. Legend: Square=reactant pair (labeled with its KEGG identifier and associated EC numbers and genes), ellipse=compound (labeled with its KEGG identifier and name).

4.8 Pathway predicted to be up-regulated in the presence of phenylalanine

4.8.1 Up-regulated genes

From the 103 genes up-regulated in the presence of phenylalanine, 29 are enzyme-coding. Table 4.7 lists the five most significantly up-regulated enzyme-coding genes with their EC numbers and reactions. ARO10 exemplifies well the difference in gene-reaction mapping between KEGG and MetaCyc: Whereas ARO10 is associated to no less than 19 reactions in KEGG (via its EC number), it is directly linked to only 4 reactions in MetaCyc.

Table 4.7: Gene-to-reactant pair mappings for genes up-regulated with phenylalanine as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
Q0045	COX1	Subunit I of cytochrome c oxidase, which is the terminal member of the mitochondrial inner membrane electron transport chain	1.9.3.1	R00081 [RP00038]
YKL165C	MCD4	Protein involved in glycosylphosphatidylinositol (GPI) anchor synthesis	2.7.-.-	R07678 [RP11388]
YDR380W	ARO10	Phenylpyruvate decarboxylase, catalyzes decarboxylation of phenylpyruvate to phenylacetaldehyde, which is the first specific step in the Ehrlich pathway	4.1.1.-	R02518 [] R06925 [RP09207] R00219 [RP04275] R04008 [RP03629] R05376 [RP04982] R04986 [RP04595] R03341 [RP02956] R04885 [RP04496] R03674 [RP03301] R05087 [RP04692] R03367 [RP02986] R05377 [RP04983] R04223 [RP03854] R02669 [RP02396] R06973 [RP10307] R04732 [RP04373] R04515 [RP04156] R02952 [RP02631] R04172 [RP03805]
YGR088W	CTT1	Cytosolic catalase T, has a role in protection from oxidative damage by hydrogen peroxide	1.11.1.6	R02670 [RP02397, RP12780] R00602 [RP00094] R00009 [RP02902]
YHR137W	ARO9	Aromatic aminotransferase II, catalyzes the first	2.6.1.57	R07396 [RP00014, RP00558] R01731 [RP01721, RP00059]

Table 4.7: Gene-to-reactant pair mappings for genes up-regulated with phenylalanine as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
		step of tryptophan, phenylalanine, and tyrosine catabolism		R00734 [RP00621, RP00014] R00694 [RP00014, RP00057] R03120 [RP00550, RP00014]

4.8.2 Predicted pathway

The predicted pathway (see Figure 4.9) consists of three components. One of them is an orphan reactant pair with its adjacent compounds (RP00038). The second component contains compounds and reactant pairs associated to tryptophan metabolism. The third component finally postulates a connection between galactose and tyrosine metabolism. This connection is very likely a false positive, as there is a lack of intermediate seeds. The tyrosine-metabolism-mapping part of the prediction contains the first steps of the tyrosine degradation pathway. Interestingly, the EC numbers of these steps (2.6.1.5 and 4.1.1.-/4.1.1.80) are the same as the EC numbers of the first steps of phenylalanine degradation as annotated in KEGG (with phenylpyruvate and phenylacetaldehyde as intermediates).

The pathway is not well supported by yeast-specific enzymes, except for its parts mapping tryptophan and tyrosine metabolism, respectively. One might assume that both are involved in the degradation of phenylalanine, which explains their up-regulation in the presence of this nitrogen source.

In the yeast-specific network, the selected genes could not be connected, because most of their associated reactant pairs were not present in this network. This discrepancy may be either due to an incomplete yeast-specific network and/or to imprecise gene-reaction mappings.

4.9 Pathway predicted to be down-regulated in the presence of phenylalanine

4.9.1 Down-regulated genes

In the presence of phenylalanine as sole nitrogen source, 24 genes are down-regulated, among them four enzymes. Table 4.8 lists the EC numbers, reactions and main reactant pairs obtained for each enzyme.

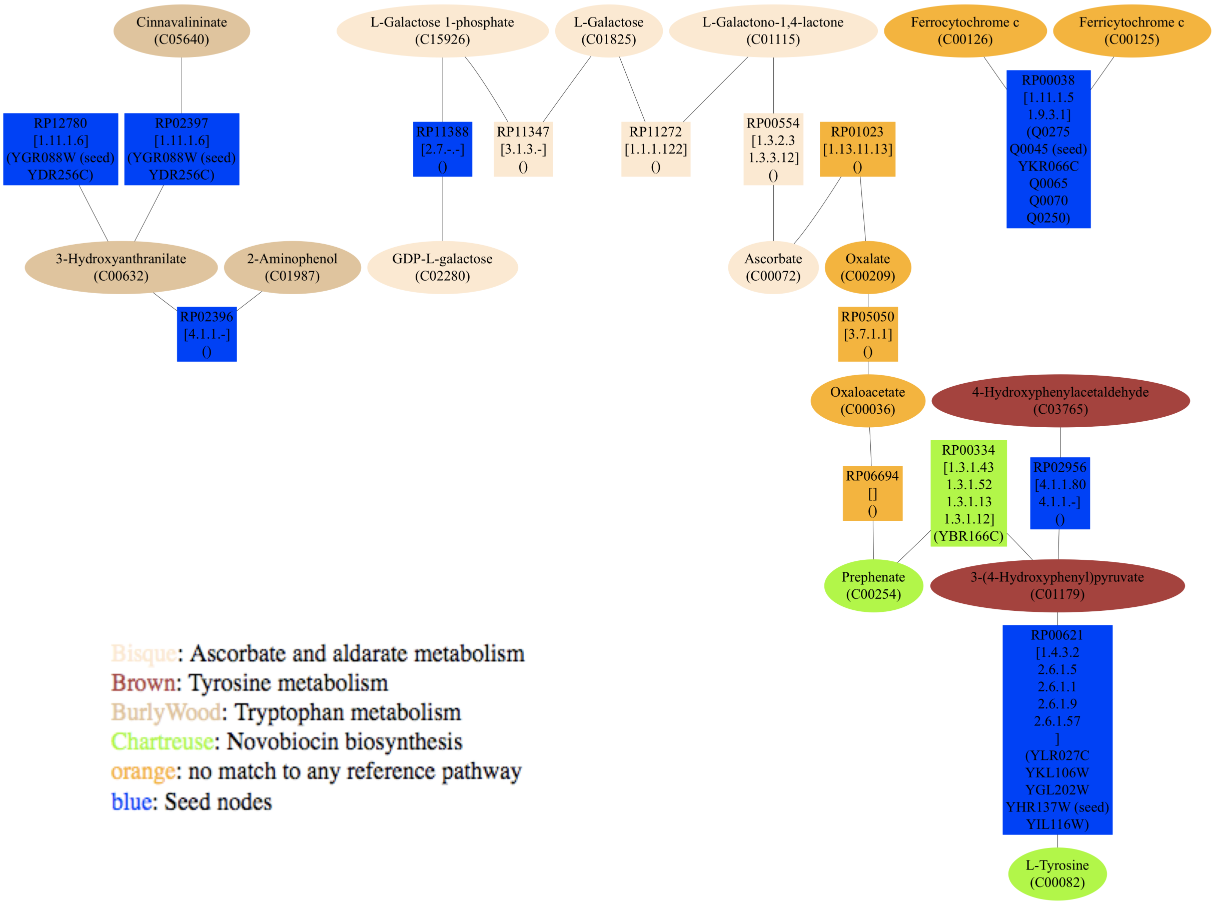

Cinnavalininate (C05640)
L-Galactose 1-phosphate (C15926)
L-Galactose (C01825)
L-Galactono-1,4-lactone (C01115)
Ferrocytochrome c (C00126)
Ferricytochrome c (C00125)
RP00038 [1.11.1.5 1.9.3.1] (Q0275 Q0045 (seed) YKR066C Q0065 Q0070 Q0250)
RP12780 [1.11.1.6] (YGR088W (seed) YDR256C)
RP02397 [1.11.1.6] (YGR088W (seed) YDR256C)
RP11388 [2.7.-.-] ()
RP11347 [3.1.3.-] ()
RP11272 [1.1.1.122] ()
RP00554 [1.3.2.3 1.3.3.12] ()
RP01023 [1.13.11.13] ()
3-Hydroxyanthranilate (C00632)
2-Aminophenol (C01987)
GDP-L-galactose (C02280)
Ascorbate (C00072)
Oxalate (C00209)
RP02396 [4.1.1.-] ()
RP05050 [3.7.1.1] ()
Oxaloacetate (C00036)
4-Hydroxyphenylacetaldehyde (C03765)
RP06694 [] ()
RP00334 [1.3.1.43 1.3.1.52 1.3.1.13 1.3.1.12] (YBR166C)
RP02956 [4.1.1.80 4.1.1.-] ()
Prephenate (C00254)
3-(4-Hydroxyphenyl)pyruvate (C01179)
RP00621 [1.4.3.2 2.6.1.5 2.6.1.1 2.6.1.9 2.6.1.57] (YLR027C YKL106W YGL202W YHR137W (seed) YIL116W)
L-Tyrosine (C00082)
Bisque: Ascorbate and aldarate metabolism
Brown: Tyrosine metabolism
BurlyWood: Tryptophan metabolism
Chartreuse: Novobiocin biosynthesis
orange: no match to any reference pathway
blue: Seed nodes

Figure 4.9: The pathway predicted from the top five enzyme-coding genes up-regulated in the presence of phenylalanine as sole nitrogen source. Legend: Square=reactant pair (labeled with its KEGG identifier and associated EC numbers and genes), ellipse=compound (labeled with its KEGG identifier and name).

Table 4.8: Gene-to-reactant pair mappings for genes down-regulated with phenylalanine as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
YIR029W	DAL2	Allantoicase, converts allantoate to urea and ureidoglycolate in the second step of allantoin degradation	3.5.3.4	R02422 [RP02197, RP02198]
YNL141W	AAH1	Adenine deaminase (adenine aminohydrolase), converts adenine to hypoxanthine	3.5.4.4	R01560 [RP01594] R06137 [] R02556 [RP02305]
YIR031C	DAL7	Malate synthase, role in allantoin degradation unknown	2.3.3.9	R00472 [RP00921, RP00007]
YGL202W	ARO8	Aromatic aminotransferase I, expression is regulated by general control of amino acid biosynthesis	2.6.1.57	R07396 [RP00014, RP00558] R01731 [RP01721, RP00059] R00734 [RP00621, RP00014] R00694 [RP00014, RP00057] R03120 [RP00550, RP00014]

4.9.2 Predicted pathway

The predicted pathway (see Figure 4.10) contains a purine degradation pathway that was also predicted for the genes down-regulated in the presence of aspartate. The purine degradation pathway ends in urate, which is predicted to be degraded to ureidoglycolate via the allantoin degradation pathway (MetaCyc identifier PWY-5697). Finally, ureidoglycolate is linked via glyoxylate, malate and oxaloacetate to aspartate.

As in the case of aspartate, one may assume that the purine degradation pathway, which is active in the presence of the bad nitrogen source urea, is not needed in the presence of the intermediate nitrogen source phenylalanine.

4.10 Pathway predicted to be up-regulated in the presence of leucine

4.10.1 Up-regulated genes

The top five genes up-regulated in the presence of leucine are mostly associated to incomplete EC numbers. Incomplete EC numbers indicate that the knowledge of the reaction mechanism is not yet elucidated. These EC numbers are problematic, because they cannot be precisely

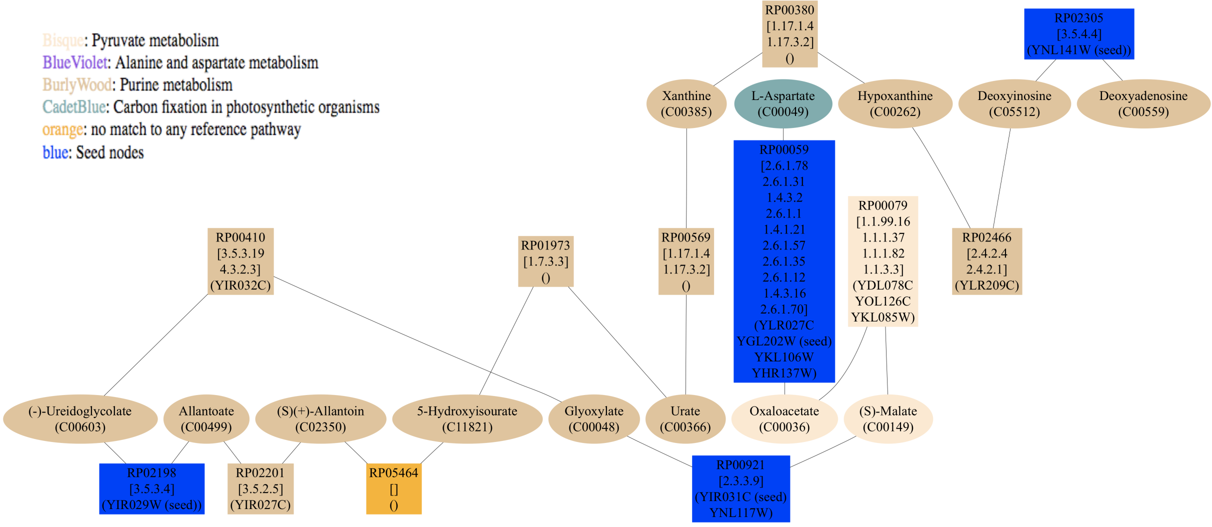

Bisque: Pyruvate metabolism
BlueViolet: Alanine and aspartate metabolism
BurlyWood: Purine metabolism
CadetBlue: Carbon fixation in photosynthetic organisms
orange: no match to any reference pathway
blue: Seed nodes
RP00380 [1.17.1.4 1.17.3.2] ()
RP02305 [3.5.4.4] (YNL141W (seed))
Xanthine (C00385)
L-Aspartate (C00049)
Hypoxanthine (C00262)
Deoxyinosine (C05512)
Deoxyadenosine (C00559)
RP00059 [2.6.1.78 2.6.1.31 1.4.3.2 2.6.1.1 1.4.1.21 2.6.1.57 2.6.1.35 2.6.1.12 1.4.3.16 2.6.1.70] (YLR027C YGL202W (seed) YKL106W YHR137W)
RP00079 [1.1.99.16 1.1.1.37 1.1.1.82 1.1.3.3] (YDL078C YOL126C YKL085W)
RP00410 [3.5.3.19 4.3.2.3] (YIR032C)
RP01973 [1.7.3.3] ()
RP00569 [1.17.1.4 1.17.3.2] ()
RP02466 [2.4.2.4 2.4.2.1] (YLR209C)
(-)-Ureidoglycolate (C00603)
Allantoate (C00499)
(S)(+)-Allantoin (C02350)
5-Hydroxyisourate (C11821)
Glyoxylate (C00048)
Urate (C00366)
Oxaloacetate (C00036)
(S)-Malate (C00149)
RP02198 [3.5.3.4] (YIR029W (seed))
RP02201 [3.5.2.5] (YIR027C)
RP05464 [] ()
RP00921 [2.3.3.9] (YIR031C (seed) YNL117W)

Figure 4.10: The pathway predicted from four enzyme-coding genes down-regulated in the presence of phenylalanine as sole nitrogen source. Legend: Square=reactant pair (labeled with its KEGG identifier and associated EC numbers and genes), ellipse=compound (labeled with its KEGG identifier and name).

mapped to reactions. KEGG nevertheless associates reactions to incomplete EC numbers. These associations will be used here, except for EC numbers where only the first level is known (e.g. 4.-.-.-). Table 4.9 lists the genes with their EC numbers and associated reactions. A special case is YIR019C (MUC1), which is associated to a complete EC number (3.2.1.3), but not to any main reactant pair. The reason is that the reactions associated to this EC number are polymeric (e.g. R01791 equation is: Dextrin + H_2O ↔ alpha-D-Glucose + Dextrin) and the metabolic networks constructed during this work do not include polymeric reactions.

Table 4.9: Gene-to-reactant pair mappings for the top five enzyme-coding genes up-regulated with leucine as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
YMR095C	SNO1	Protein of unconfirmed function, involved in pyridoxine metabolism	2.6.-.-	R07386 [RP10955, RP11005]
YDR380W	ARO10	Phenylpyruvate decarboxylase, catalyzes decarboxylation of phenylpyruvate to phenylacetaldehyde, which is the first specific step in the Ehrlich pathway	4.1.1.-	R02518 [] R06925 [RP09207] R00219 [RP04275] R04008 [RP03629] R05376 [RP04982] R04986 [RP04595] R03341 [RP02956] R04885 [RP04496] R03674 [RP03301] R05087 [RP04692] R03367 [RP02986]

Table 4.9: Gene-to-reactant pair mappings for the top five enzyme-coding genes up-regulated with leucine as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
				R05377 [RP04983] R04223 [RP03854] R02669 [RP02396] R06973 [RP10307] R04732 [RP04373] R04515 [RP04156] R02952 [RP02631] R04172 [RP03805]
YMR096W	SNZ1	Protein involved in vitamin B6 biosynthesis	4.-.-.-	-
YIR019C	MUC1	GPI-anchored cell surface glycoprotein (flocculin) required for pseudohyphal formation, invasive growth, flocculation, and biofilms	3.2.1.3	-
YHR137W	ARO9	Aromatic aminotransferase II, catalyzes the first step of tryptophan, phenylalanine, and tyrosine catabolism	2.6.1.57	R07396 [RP00014, RP00558] R01731 [RP01721, RP00059] R00734 [RP00621, RP00014] R00694 [RP00014, RP00057] R03120 [RP00550, RP00014]

4.10.2 Predicted pathway

The pathway predicted for four genes most significantly up-regulated in the presence of leucine as sole nitrogen source is shown in Figure 4.11. It consists of one component, a part of which maps to tyrosine metabolism and contains the first steps of the tyrosine degradation pathway. 4-Hydroxyphenylacetate is connected to choline via Penicillin G, which very likely represents a false positive.

The predicted pathway suggests an up-regulation of tyrosine degradation in the presence of leucine as sole nitrogen source.

Unfortunately, the prediction could not be repeated in the yeast-specific network, because it did not contain enough seed reactant pairs.

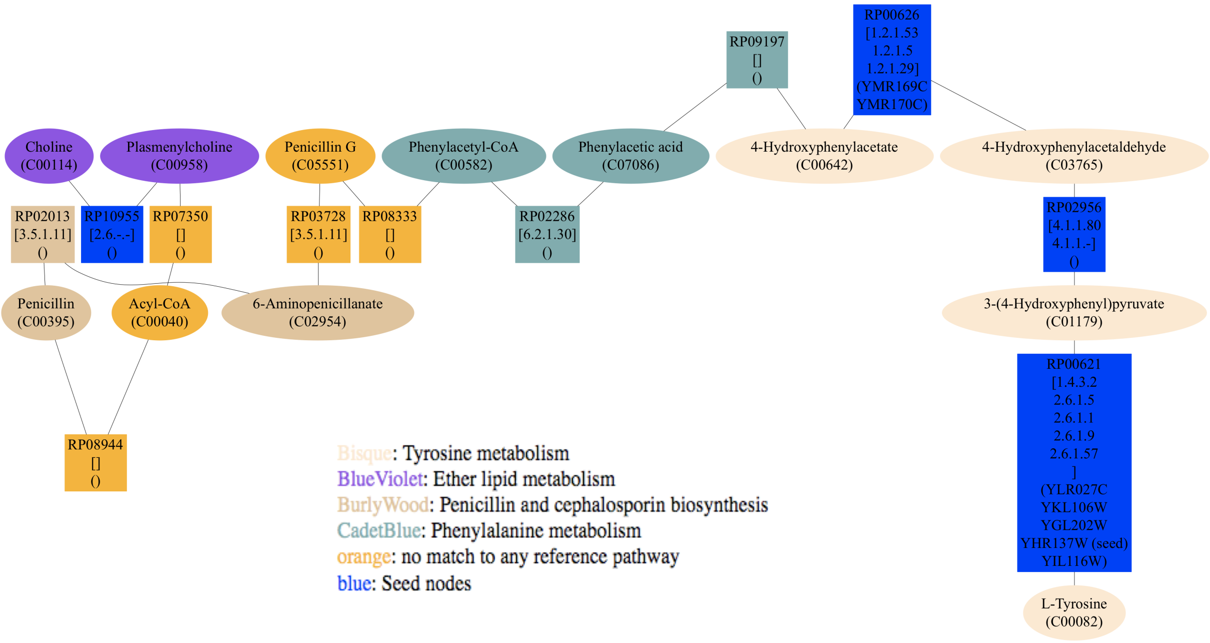


Figure 4.11: The pathway predicted from the top five enzyme-coding genes up-regulated in the presence of leucine as sole nitrogen source. Legend: Square=reactant pair (labeled with its KEGG identifier and associated EC numbers and genes), ellipse=compound (labeled with its KEGG identifier and name).

4.11 Pathway predicted to be down-regulated in the presence of leucine

4.11.1 Down-regulated genes

26 out of 5,690 *S. cerevisiae* genes were identified to be significantly down-regulated in the presence of leucine, nine of them enzyme-coding. Table 4.10 lists the top five enzyme-coding genes together with their EC numbers and reactions. As for aspartate as sole nitrogen source, two malate synthases are down-regulated.

Table 4.10: Gene-to-reactant pair mappings for the top five enzyme-coding genes down-regulated with leucine as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
YIR029W	DAL2	Allantoicase, converts allantoate to urea and ureidoglycolate in the second step of allantoin degradation	3.5.3.4	R02422 [RP02197, RP02198]
YNL141W	AAH1	Adenine deaminase (adenine aminohydrolase), converts	3.5.4.4	R01560 [RP01594] R06137 []

Table 4.10: Gene-to-reactant pair mappings for the top five enzyme-coding genes down-regulated with leucine as sole nitrogen source

Gene identifier	Gene name	Gene description	EC numbers of gene	Reactions and main reactant pairs of EC number
		adenine to hypoxanthine		R02556 [RP02305]
YNL117W	MLS1	Malate synthase, enzyme of the glyoxylate cycle, involved in utilization of non-fermentable carbon sources	2.3.3.9	R00472 [RP00921, RP00007]
YLR142W	PUT1	Proline oxidase, nuclear-encoded mitochondrial protein involved in utilization of proline as sole nitrogen source	1.5.99.8	R01253 [RP00228] R05051 [RP04655]
YIR031C	DAL7	Malate synthase, role in allantoin degradation unknown	2.3.3.9	R00472 [RP00921, RP00007]

4.11.2 Predicted pathway

The prediction (see Figure 4.12) suggests that a proline degradation pathway (that proceeds via L-erythro-4-hydroxyglutamate and ends in the glyoxylate cycle) is down-regulated in the presence of leucine. The pathway also contains the purine and allantoin degradation pathways predicted to be down-regulated in the presence of aspartate.

Leucine clustered with slow-growth nitrogen sources, but is stated to support quicker growth than the other group B nitrogen sources [63]. One may hypothesize that the purine and proline degradation pathways, which might generate extra nitrogen in a nitrogen-starved cell, are not needed in the presence of leucine. Interestingly, the down-regulation of a proline degradation pathway was not predicted for any other group B nitrogen source, whereas the down-regulation of the purine degradation pathway is predicted for isoleucine and methionine as well (for threonine, no results were obtained).

4.12 Comparison of nitrogen sources based on predicted pathways

So far, the results for three nitrogen sources (a good, a bad and an intermediate one) were presented. When inspecting the predictions for the other nitrogen sources, it becomes apparent that the same pathways, such as proline, allantoin and purine degradation, occur in several conditions.

Thus, nitrogen sources could be grouped based on the similarity of the metabolic pathways they activate or suppress.

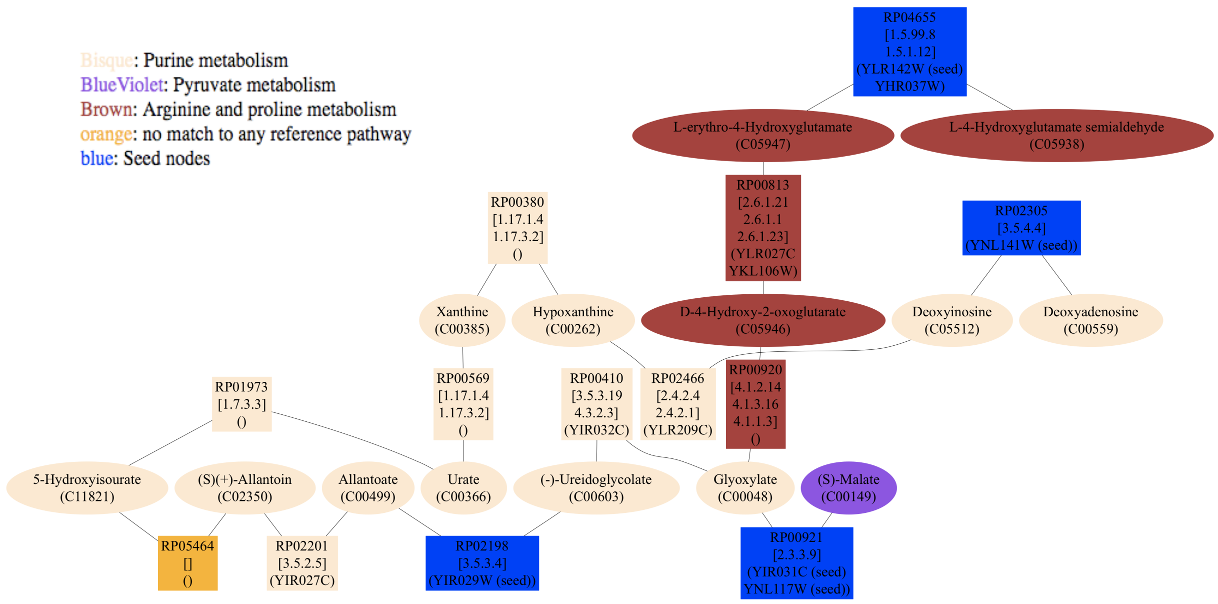

Bisque: Purine metabolism
BlueViolet: Pyruvate metabolism
Brown: Arginine and proline metabolism
orange: no match to any reference pathway
blue: Seed nodes
RP04655 [1.5.99.8 1.5.1.12] (YLR142W (seed) YHR037W)
L-erythro-4-Hydroxyglutamate (C05947)
L-4-Hydroxyglutamate semialdehyde (C05938)
RP00380 [1.17.1.4 1.17.3.2] ()
RP00813 [2.6.1.21 2.6.1.1 2.6.1.23] (YLR027C YKL106W)
RP02305 [3.5.4.4] (YNL141W (seed))
Xanthine (C00385)
Hypoxanthine (C00262)
D-4-Hydroxy-2-oxoglutarate (C05946)
Deoxyinosine (C05512)
Deoxyadenosine (C00559)
RP01973 [1.7.3.3] ()
RP00569 [1.17.1.4 1.17.3.2] ()
RP00410 [3.5.3.19 4.3.2.3] (YIR032C)
RP02466 [2.4.2.4 2.4.2.1] (YLR209C)
RP00920 [4.1.2.14 4.1.3.16 4.1.1.3] ()
5-Hydroxyisourate (C11821)
(S)(+)-Allantoin (C02350)
Allantoate (C00499)
Urate (C00366)
(-)-Ureidoglycolate (C00603)
Glyoxylate (C00048)
(S)-Malate (C00149)
RP05464 [] ()
RP02201 [3.5.2.5] (YIR027C)
RP02198 [3.5.3.4] (YIR029W (seed))
RP00921 [2.3.3.9] (YIR031C (seed) YNL117W (seed))

Figure 4.12: The pathway predicted from the top five enzyme-coding genes down-regulated in the presence of leucine as sole nitrogen source. Legend: Square=reactant pair (labeled with its KEGG identifier and associated EC numbers and genes), ellipse=compound (labeled with its KEGG identifier and name).

More precisely, the distance between two nitrogen sources i and j can be defined as

$$dist_{i,j} = 1 - Jaccard(i, j) \tag{4.1}$$

Given node set A from the pathway predicted for nitrogen source i and node set B from the pathway predicted for nitrogen source j, the Jaccard coefficient is defined as:

$$Jaccard = \frac{A \cap B}{A \cup B} \tag{4.2}$$

Orphan reactant pairs (and their adjacent compounds) are not taken into account, to avoid a bias due to many shared nodes for a broad-specificity enzyme.

Once the distance between two nitrogen sources is defined, the nitrogen sources can be clustered as shown in Figure 4.13. Clusters were obtained using function "heatmap" with method "ward" in R, other cluster algorithms yield similar results. For comparison, the clustering of nitrogen sources based on gene expression ratios is shown in Figure 4.1.

When analyzing the pathway-based clustering, the following can be noted:

- In general, pathways are very dissimilar.
- Pathways are roughly divided into up- and down-regulated.
- Notable clusters (with distances below 0.6) are formed by: (1) pathways up-regulated in the presence of aspartate, glutamate and proline, (2) pathways up-regulated in the presence of alanine and ammonium, (3) pathways up-regulated in the presence of glutamine

and valine, (4) pathways down-regulated in the presence of aspartate, serine, isoleucine, arginine, leucine and phenylalanine, (5) pathways down-regulated in the presence of ammonium and asparagine and (6) pathways down-regulated in the presence of tryptophan and tyrosine.

- The clustering is different to the one obtained from the gene expression ratio matrix. This may be due to the separation into up- and down-regulated pathways and the neglect of non-enzyme-coding genes. Notably, good and bad nitrogen sources were not separated as clearly as in Figure 4.1.
- Nitrogen sources with intersecting up-regulated pathways do not necessarily have intersecting down-regulated pathways.

4.13 Discussion and conclusion

The evaluation of two seed reaction grouping strategies showed that EC groups perform slightly better than reaction groups. Grouping seed reactions on the gene level was not tested, but might be useful to predict pathways from genes that can be directly linked to reactions.

Interestingly, some good and bad nitrogen sources cluster together, mainly because of the allantoin and purine degradation pathways down-regulated in their presence (e.g. for aspartate, phenylalanine and leucine). The pathways are down-regulated with respect to urea as sole nitrogen source and may therefore be specific to this reference nitrogen source rather than the tested nitrogen sources.

The up-regulated pathways are more variable and may be specific to the investigated nitrogen sources. In a few cases, a part of the up-regulated pathway corresponds to a reference pathway degrading the given nitrogen source (phenylalanine, GABA). In other cases (aspartate, asparagine, ammonium) storage compound formation is up-regulated.

During prediction of pathways from up- and down-regulated gene groups, several difficulties were encountered: First, it is not obvious how many of the differentially expressed genes in each group should serve as input to pathway prediction. A conservative threshold (E value set to one) resulted in quite large gene groups and pathways predicted for them were hard to visualize and to interpret. One could either divide those pathways in smaller units using a graph cluster algorithm or, as done in this chapter, keep only the genes with lowest E values. In the latter case, a threshold is required, which was set arbitrarily to five. Second, gene-to-reaction mapping is ambiguous in KEGG. Genes are not directly linked to reactions (apart from the links hidden in the KGML files that are not accessible via the KEGG API or web interface), thus genes have to be linked to reactions via EC numbers, which introduces false positive reactions. EC numbers are sometimes incomplete (e.g. YMR095C with EC number 2.6.-.-), or describe broad-specificity enzymes associated to a large number of reactions (e.g. YGL256W alias ADH4). Third, there is a tradeoff in network choice. On the one hand, a species-specific metabolic network cannot discover varieties of pathways yet unknown in this species. On the other hand, a network that is too generic introduces many false positives. Fourth, KEGG contains generic compounds such as alpha-amino acid, whose treatment is not

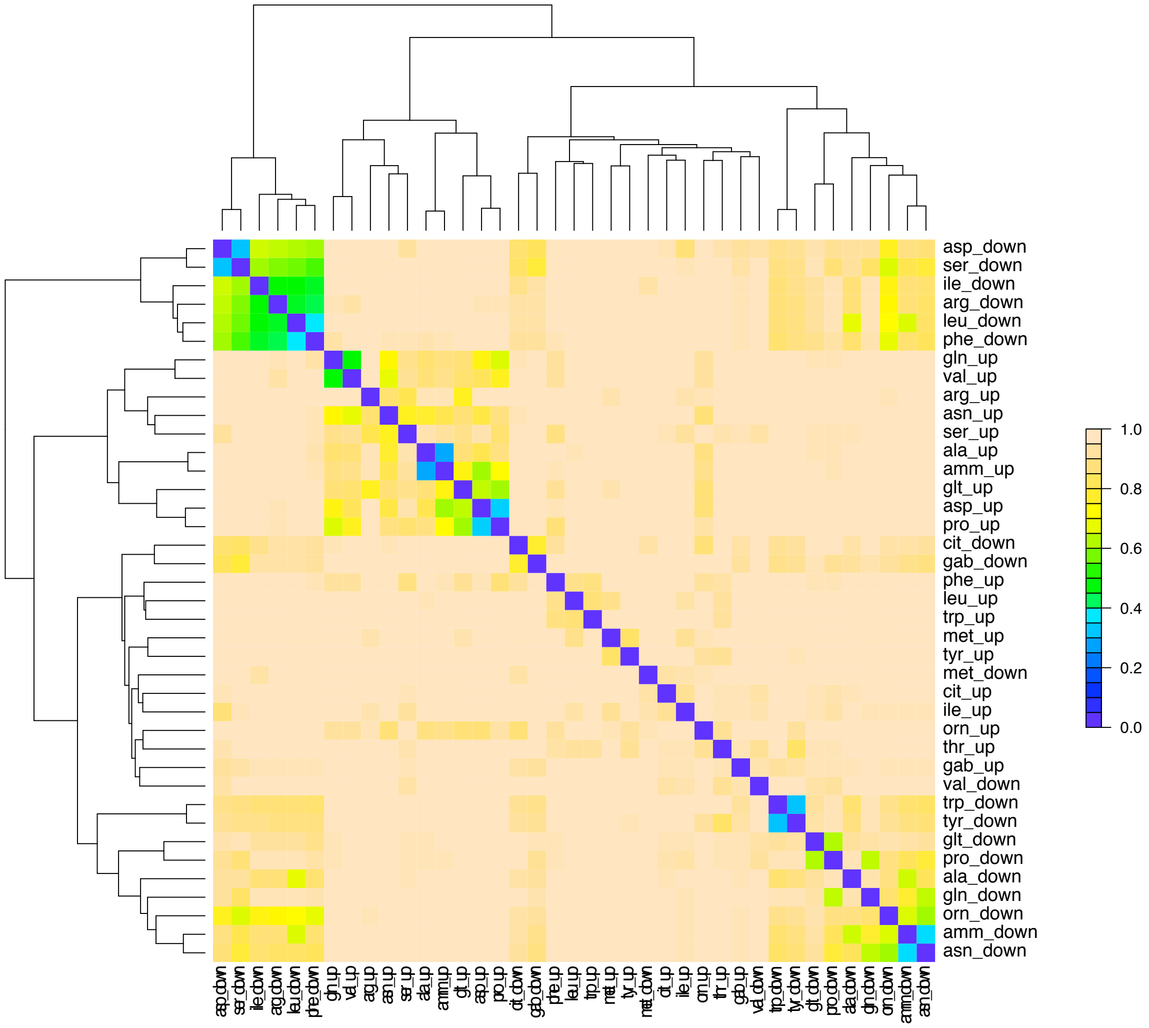


Figure 4.13: Heat map of nitrogen source distances. The columns of the matrix are re-arranged according to the outcome of the "ward" cluster algorithm. The distance ranges between zero (identical pathways) and one (intersection of pathways is zero).

obvious. They are useful to describe broad-specificity reactions, but may introduce dubious connections.

Thus, errors are introduced on several levels: during the identification of significantly overexpressed genes, during the mapping of genes to reactions and finally during pathway prediction. To reduce errors in future applications of pathway prediction, one may consider other databases than KEGG to map genes to reactions. For instance, MetaCyc offers more precise gene-to-reaction mappings and in addition cross-references its reactions to KEGG. Several taxonomic levels could be tested to construct a metabolic network with sufficient sensitivity (new pathways can be detected) and specificity (most reactions are supported by species-specific enzymes). In general, species-specific KEGG metabolic networks are of low quality. One might therefore consider the high-quality metabolic reconstruction of yeast metabolism recently described in [141].

Overall, this chapter highlights the many problems encountered when interpreting microarray data on the pathway level and the presented results should be considered as preliminary.

5 Stoichiometric versus non-stoichiometric pathway prediction

Presented letter to the editor:
K. Faust, D. Croes and J. van Helden
In response to "Can sugars be produced from fatty acids? A test case for pathway analysis tools"
Bioinformatics, vol. 25, pp. 3202-3205, 2009.

5.1 Introduction

The question can be raised whether the stoichiometry of compounds should be taken into account during pathway prediction. This chapter deals with this question.

First, taking into account the stoichiometry of pathways means to stoichiometrically balance the metabolic pathway. Merely labeling the reactions of a pathway with their stoichiometries can easily be done by querying a metabolic database and is not a point of discussion.

In a *stoichiometrically balanced* pathway, as many compounds are produced as consumed and reactions can only occur if their substrates are present in sufficient numbers.

The prediction approaches presented in chapter 2 and 3 do not stoichiometrically balance the predicted pathway. However, several prediction approaches presented in the Introduction (such as elementary mode analysis and the constraint-based approaches by Planes & Beasley, see section 1.10.3) are designed to predict balanced pathways only. Figure 5.1 shows the balanced and non-balanced part of a reference pathway. In the reference pathways listed in MetaCyc and aMAZE, side compounds are usually not balanced. A pathway prediction approach that predicts balanced pathways is thus not capable to predict these reference pathways, except if it excludes certain compounds from the balancing condition (called *external compounds* in EM analysis). Thus, side compounds need to be defined before pathway prediction, which poses the difficulties mentioned in the Introduction: (1) It is not clear beforehand which compounds are side compounds (2) Pathways synthesizing or degrading the side compounds are excluded together with the side compounds.

On the other hand, if stoichiometry is neglected in pathway prediction, biochemically irrelevant pathways may result. A study case illustrating this was recently presented by de Figueiredo et al. [38] and concerns the synthesis of sugars from fatty acids. Fatty acids are essentially broken down to acetyl-CoA molecules. Several organisms (e.g. *E. coli* and

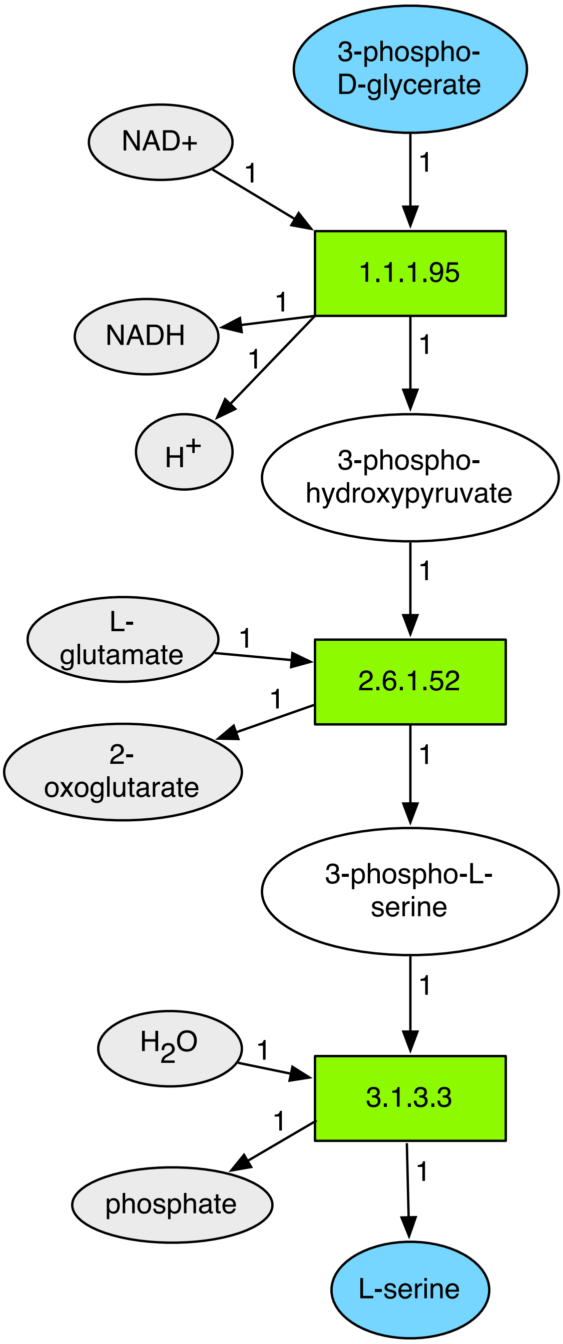


Figure 5.1: The serine biosynthesis pathway as annotated in MetaCyc (identifier: SERSYN-PWY) is shown. Each reaction is displayed with all its substrates and products. Each arc connecting a compound and a reaction is labeled with the *stoichiometric coefficient*, i.e. the number of units of the compound consumed or produced by the reaction. In this pathway, only the white compounds are balanced, that is they are produced and consumed in equal numbers. The blue compounds (start and end compound of the pathway) as well as the grey compounds (side compounds) are not balanced. An approach that stoichiometrically balances pathways can predict the serine biosynthesis pathway only if it excludes certain compounds from the balancing condition. In EM analysis, such non-balanced compounds are called *external compounds*.

plants) can synthesize sugars (such as glucose) from acetyl-CoA. In the TCA cycle, acetyl-CoA is converted into oxaloacetate, which can enter the glucose-yielding gluconeogenesis pathway. However, in the TCA cycle, acetyl-CoA is entirely degraded to CO_2. Without stoichiometrically balancing the pathway, a prediction method could falsely predict that sugars are synthesized from fatty acids via the TCA cycle. Figure 5.2 illustrates this study case.

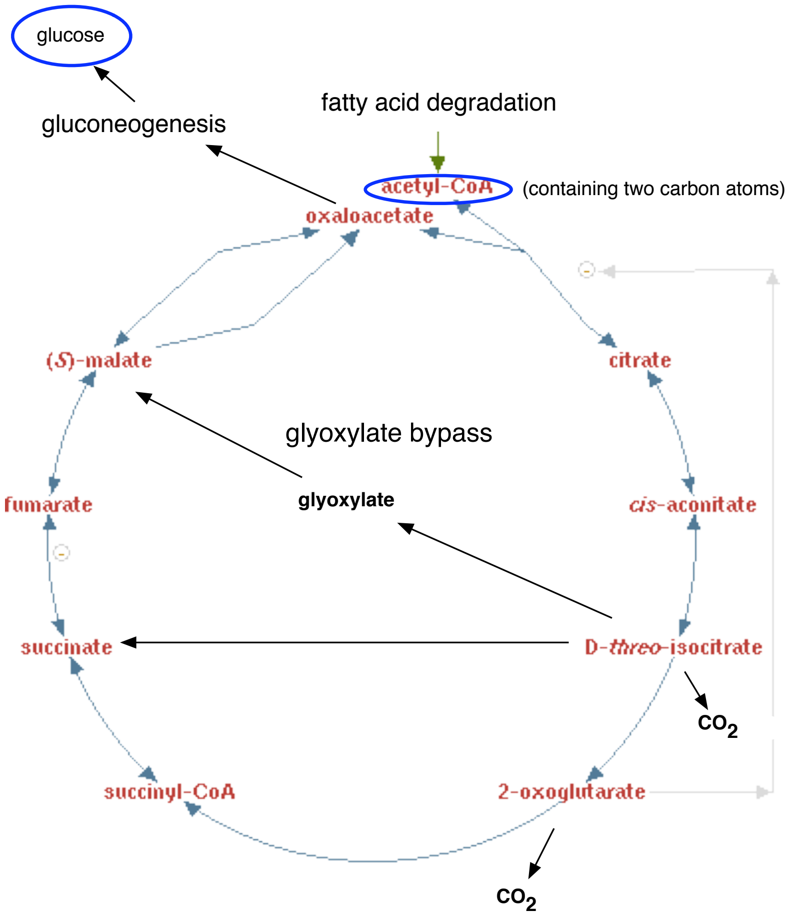


Figure 5.2: Illustration of the study case presented by de Figueiredo et al. [38]. Acetyl-CoA is converted into oxaloacetate via the TCA cycle. Oxaloacetate in turn can enter the gluconeogenesis pathway, which yields glucose. However, the TCA cycle degrades acetyl-CoA entirely to carbon dioxide. It is therefore not possible to synthesize glucose from acetyl-CoA via this cycle. Since path finding does not balance the pathway, it could nevertheless predict a pathway that falsely converts the start compound acetyl-CoA into the end compound glucose via this cycle. A bypass such as the glyoxylate cycle is needed to enable the net synthesis of glucose from acetyl-CoA. Image sources: The pathway image was taken from MetaCyc. Text and arrows in black as well as blue ellipses were added by the author.

5.2 Treatment of the fatty-acid-to-sugar study case without stoichiometric balance

It is worthwhile exploring whether the fatty-acid-to-sugar study case and similar cases can be treated correctly without stoichiometrically balancing the predicted pathway.

5.2.1 Atom tracing

First, one may ask whether the study case can be treated correctly by atom tracing. In order to check whether a net production via the TCA cycle is possible, atom tracing methods have to trace not only the fatty acid atoms to the desired target compound (the sugar), but also through the TCA cycle. Thus, the atom tracing method needs to follow all the atoms of an input compound through the network and in addition to keep track of cycling atoms (such as those in the TCA cycle) by counting atoms. This amounts to stoichiometrically balancing the compounds of a pathway.

5.2.2 Cycle treatment

If one assumes that all problematic cases are caused by a special kind of metabolic cycle, one can attempt to identify all such cycles in the metabolic network. If all cycles without net production of compounds could be identified prior to pathway prediction, path finding could be forbidden to reach a target compound via such a cycle. Whether simple assumptions like this one are sufficient to remedy path finding is an open question.

5.3 Assumptions of stoichiometrically balanced and unbalanced pathways

From what has been said so far, we may conclude that pathway prediction methods should always stoichiometrically balance a predicted pathway. There is however another turn to the story.

When stoichiometrically balancing a pathway, it is assumed that the metabolic network is completely known, otherwise compounds cannot be balanced correctly (as unknown reactions may consume or produce them). For stoichiometrically unbalanced pathways, this assumption is not made, but it is assumed that compounds are always available in sufficient numbers for the pathway to proceed.

Concerning the study case, non-stoichiometric pathway prediction assumes that another part of the network will generate oxaloacetate from the source compound by other routes than the TCA cycle (which may or may not be the case), whereas stoichiometric pathway prediction assumes that the network (where oxaloacetate is generated from the source compound only via the TCA cycle) is complete (which may or may not be the case).

Thus, stoichiometric pathway prediction is the method of choice in case a well curated, complete metabolic network is available. For incomplete metabolic networks, pathway prediction with stoichiometry is not superior to pathway prediction without stoichiometry, as both make (possibly unjustified) assumptions in this case.

5.4 Stoichiometry and incomplete metabolic networks

The less is known about the metabolism of an organism, the less accurate will be predictions from both stoichiometric and non-stoichiometric approaches. However, stoichiometric approaches are more strongly affected by incomplete networks.

In contrast to non-stoichiometric approaches, stoichiometric approaches may predict the absence of a number of pathways, because balancing reactions for certain compounds are absent from the network. These pathways will be false negatives in case balancing reactions are present in the organism, but have been missed during construction of the network.

For instance, consider the fatty-acid-to-sugar study case and assume that the organism in question possesses the glyoxylate cycle. In this case, a stoichiometric approach, based on the incomplete metabolic network, would wrongly predict that fatty acids cannot be converted into sugars, whereas a non-stoichiometric approach (which assumes that oxaloacetate is replenished by some other part of metabolism) would probably predict a pathway close to the true pathway (where the true pathway is determined by experimentally tracing carbon atoms from fatty acids to sugar).

5.5 EM analysis versus path finding

There are different ways of stoichiometrically balancing a pathway, one of which is EM analysis (see Introduction, section 1.10.3).

In the same article that presented the fatty-acid-to-sugar study case, de Figueiredo and coauthors also compared EM analysis with path finding (for path finding, see section 1.10.3). This comparison had several weaknesses, which prompted us to respond in a comment. In this comment, we not only point out the weaknesses of the comparison of EM analysis and path finding, we also discuss strengths and weaknesses of both approaches.

Some of the weaknesses of EM analysis listed in our comment have meanwhile been addressed. This concerns the following weaknesses:

- EM analysis can only be applied to small networks: Recently, methods have been published which allow the application of EM analysis to genome-scale networks [157, 85, 37].
- In large networks, EM analysis enumerates millions of EMs: A strategy to restrict the number of EMs has been published recently, which consist in ranking EMs according to their length [37].

Apart from the weaknesses of EM analysis mentioned in the comment, there are additional weaknesses listed below:

- EM analysis cannot deal with parallel pathways or isoenzymes, which have to be removed from the metabolic network.

- EM analysis assumes that the compound concentrations remain approximately constant at an appropriate time scale. This is motivated by stating that compound pools rapidly reach a steady state (within seconds). However, recent measurements in *S. cerevisiae* have shown that there are slow compound pools [118]. Thus, the steady state assumption is not valid for all compounds to the same extend. Slow compounds could be considered as external compounds, but without measurements, it is not clear which are the slow compounds.

5.6 Alternative stoichiometric approaches

The stoichiometric pathway prediction approach by Planes and Beasley [133] (see Introduction, section 1.10.3) does not suffer from many of the drawbacks of EM analysis (e.g. steady state assumption, assignment of external compounds, millions of EMs) and copes with large networks. In contrast to many path finding approaches, it also deals well with cycles. Moreover, this approach has been systematically evaluated on reference pathways from *E. coli*. Thus, for the prediction of stoichiometrically balanced pathways given two seed nodes, this approach may be more appropriate than EM analysis.

5.7 Stoichiometric balance and feasibility of metabolic pathways

One may ask whether metabolic pathways should not only be balanced, but also be feasible. In other words, should a pathway produce and consume all its compounds in equal numbers and generate all its compounds from the given seed nodes? The difference becomes clear when we consider the pair NADPH/NADP. In a balanced pathway, as many units of NADPH and NADP are produced as consumed. In a feasible pathway, NADPH and NADP have to be synthesized by the pathway (or to be excluded from the feasibility condition). Chemical organizations (see Introduction, section 1.7.6) are an example for systems that generate all their compounds via stoichiometrically balanced pathways. Whether or not pathways should be balanced and feasible largely depends on the application in mind. One such application may be the construction of in vitro pathways that synthesize a desired compound [73].

5.8 Conclusion

From the comparison of stoichiometric versus non-stoichiometric prediction approaches, one can conclude that each is appropriate in a different situation. Non-stoichiometric approaches should be applied when predicting pathways from a large, possibly incomplete network, which may or may not be organism-specific. In contrast, stoichiometric approaches should be applied when predicting pathways from a well-curated, organism-specific network. Of course, one of the goals of pathway prediction, namely the discovery of new pathways, is less interesting in a well-known network.

Bioinformatics Advance Access published September 23, 2009

LETTER TO THE EDITOR

In response to “Can sugars be produced from fatty acids? A test case for pathway analysis tools”

Karoline Faust*, Didier Croes and Jacques van Helden

Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe). Université Libre de Bruxelles, Campus Plaine,

CP 263. Bld du Triomphe. B-1050 Bruxelles, Belgium.

Associate Editor: Prof. Alfonso Valencia

ABSTRACT

Motivation: In their article entitled “Can sugars be produced from fatty acids? A test case for pathway analysis tools” de Figueiredo and co-authors assess the performance of three pathway prediction tools (METATOOL, PathFinding and Pathway Hunter Tool) using the synthesis of glucose-6-phosphate (G6P) from acetyl-CoA in humans as a test case (de Figueiredo, et al., 2008). We think that this article is biased for three reasons: (1) The metabolic networks used as input for the respective tools were of very different sizes; (2) the “assessment” is restricted to two study cases; (3) developers are inherently more skilled to use their own tools than those developed by other people.

We extended the analyses led by de Figueiredo and clearly show that the apparent superior performance of their tool (METATOOL) is partly due to the differences in input network sizes. We also see a conceptual problem in the comparison of tools that serve different purposes. In our opinion, metabolic path finding and elementary mode analysis are answering different biological questions, and should be considered as complementary rather than competitive approaches.

1 INTRODUCTION

The CO_2-releasing reactions of the Krebs cycle need to be bypassed in order to synthesize G6P (a sugar) from acetyl-CoA via this cycle. Such a bypass is the glyoxylate cycle, which is active in many organisms capable of growing on fatty acids (such as certain plants and bacteria), but absent in humans (Berg, et al., 2002). Based on this study case, de Figueiredo and colleagues compare the following tools:

(1) METATOOL 5.0 (von Kamp and Schuster, 2006), which computes the elementary modes (EM) of a metabolic network. An elementary mode is defined as the “minimal set of enzymes that could operate at steady state” (Pfeiffer, et al., 1999). The steady state condition requires the stoichiometric balance (production and consumption in equal numbers) of all compounds that are not external to the metabolic network.

(2) PathFinding (Croes, et al., 2005; Croes, et al., 2006) is based on a k-shortest paths algorithm (depth-first search). It circumvents the problem of highly connected compounds (van Helden, et al., 2002) by assigning weights to compounds according to their degree in the metabolic network.

(3) Pathway Hunter Tool (PHT) (Rahman, et al., 2004) is also based on a k-shortest paths algorithm (depth-first search). It relies on the chemical similarity of substrate-product pairs to avoid side compounds.

We will refer to pathway prediction tools relying mainly on network topology and ignoring stoichiometry (such as PathFinding and PHT) as path finding tools.

In this letter, we do not contest the main conclusion of the authors that only METATOOL deals correctly with the selected test case. Indeed, this test case illustrates well some problems that can arise when the stoichiometry is ignored in pathway prediction. However, we question the evaluation procedure, and therefore the generality of their conclusion. We also give a different view on the respective strengths and weaknesses of EM-based versus path finding approaches.

2 BIASES IN THE EVALUATION PROCEDURE

2.1 Tools should be fed with the same input network

Instead of supplying all three tools with the same metabolic network, each of them was tested with a different input metabolic network. METATOOL was fed with a network consisting of 29 carefully selected reactions involved in central metabolism. PHT was launched on a human-specific network obtained from KEGG (Kanehisa, et al., 2008) (v.39.0) containing 1,492 reactions and 1,490 compounds. PathFinding was tested with the complete KEGG small molecule metabolic network, which comprised 5,985 reactions and 5,082 compounds (Croes, et al., 2005), and its results were a posteriori filtered to keep only paths present in the human-specific KEGG network. Considering this difference in input network size, it is not surprising that METATOOL comes closest to the expected result.

To demonstrate that network size does matter in such a comparison, we ran PathFinding on the same input network as given to METATOOL. For this, we constructed two bipartite networks from the 29 reactions (including the glyoxylate cycle) used by de Figueiredo, one taking into account reaction directionality and another one treating all reactions as reversible. Compounds were assigned a weight equal to their degree in the complete small molecule metabolic network (KEGG LIGAND version 44.0), as done in (Croes, et al., 2006). Given these small networks, PathFinding returned the glyoxylate cycle and gluconeogenesis pathway as top-ranking paths (supplementary material), which is not the case if it is run on the full KEGG network as done by de Figueiredo.

2.2 Tools should be assessed on a representative number of study cases

The comparison presented by de Figueiredo et al. is also questionable because of its restriction to two carefully selected study cases. This results in a biased view on the quality of the compared tools. Path finding tools are particularly weak in discovering paths situated in the highly connected core of large metabolic networks (e.g. glycolysis, TCA cycle, etc.). In contrast, these pathways have been studied with EM analysis before, e.g. (Carlson, et al., 2004; Schwartz and Kanehisa, 2006), which shows that the study case was not selected at random. A fair comparison should be based either

*To whom correspondence should be addressed.

on the analysis of all available pathways, or on a representative and unbiased selection of them.

2.3 Tool comparisons should be carried out by neutral assessors

Lastly, we would like to point out that the comparison of the three pathway prediction tools has been carried out by authors involved in the development of one of these tools (namely METATOOL). This induces an inherent bias in the comparison, as a developer is more skilled in the use of his/her own tools than in those developed by other people. A fairer procedure would be to follow a CASP-like protocol (Moult, et al., 2007), where developers would use their own tools on a set of test cases, and the evaluation of the results would be carried out by an independent committee. However, in our opinion it is not meaningful to compare pathway prediction performance of EM analysis and path finding approaches, since they differ in their definition of a metabolic pathway and have been designed for different tasks.

3 DISCUSSION

The main difference between EM analysis and path finding approaches lies in their definition of a metabolic pathway. Whereas EM analysis considers as valid pathways only those metabolic sub-networks where all internal compounds are stoichiometrically balanced, path finding approaches do not impose any stoichiometric constraints. In our opinion, this fundamental distinction in the definition of a metabolic pathway leads to different advantages and disadvantages of both types of tools for pathway prediction, which are summarized below.

Advantages of EM-based tools: (1) They predict pathways whose internal compounds are balanced; (2) consequently, they also predict the pathway stoichiometry; (3) they can treat cyclic pathways.

Disadvantages of EM-based tools: (1) They require the user to decide for each compound of a given metabolic network whether it is internal or external to the system. In case the assignment of a compound as internal is faulty, EM analysis might fail to detect biochemically valid pathways; (2) EM-based tools cannot predict pathways that do not fit into their definition of a pathway. Planes and Beasley (Planes and Beasley, 2008) listed a number of classical pathways that are missed by EM-based tools for this reason (among them arginine biosynthesis and pentose phosphate salvage pathway); (3) combinatorial explosion limits the size of the input network that EM-based tools can treat. For instance, METATOOL 5.0 computed 2,450,787 EMs for a metabolic network of *E. coli* consisting of 112 reactions and 89 internal compounds within 87 minutes (von Kamp and Schuster, 2006); (4) predicted EMs are not ranked, which makes it hard to inspect EM analysis results for large metabolic networks (e.g. networks with more than 100 nodes). However, they may be ranked according to molar yield of a compound of interest (Trinh, et al., 2009) or on the basis of experimentally measured fluxes (Schwartz and Kanehisa, 2006); (5) the steady state constraint might not be appropriate for the metabolic network under study. Indeed, an organism may live in a rapidly changing environment, where the utilization of some substrates will modify their external concentration, which will be compensated by the activation or repression of other pathways. The steady-state constraint on the models seems essentially valid when experiments are performed in perfectly controlled conditions such as a chemostat, so that the system is really in steady state. In such conditions, it is crucial to carefully choose the set of reactions included in the network in order to ensure that the derived mathematical model can reach a steady state. For example, Teusink et al. modified their model of glycolysis because numerical simulations performed with experimentally measured metabolite concentrations had shown that this model failed to reach a steady state (Teusink, et al., 2000).

Path finding tools have the following advantages: (1) They can deal with large input networks (e.g. all currently known compounds and reactions); (2) they do not require the partition of compounds into internal and external (though some path finding tools need additional knowledge such as the compound structure); (3) they allow the integration of contextual information. For instance, weights can express the probability of a reaction to occur in the organism under study; (4) they rank predicted paths according to a certain criterion such as path weight or degree of similarity between compounds in subsequent steps.

Disadvantages of path finding tools: (1) They do not take into account stoichiometry and therefore do not guarantee that predicted paths allow the net synthesis of a given end compound; (2) consequently, they cannot predict the stoichiometry of paths; (3) they cannot find cyclic pathways (trivial cycles excepted) or pathways in which the same enzymes act repeatedly on a growing chain (e.g. in fatty acid elongation). However, they can predict parts of these pathways; (4) path finding tools have difficulties to correctly predict pathways located in the densely connected central metabolism, such as glycolysis. In this region, currently employed criteria such as weight or compound similarity are not sufficient to identify relevant pathways in large networks; (5) most path finding tools require parameter tuning with respect to the given input network (node weights, cut-off on the number of paths requested).

For a more in-depth discussion of different pathway prediction approaches we recommend (Planes and Beasley, 2008).

Not only do EM-based and path finding tools differ in their strengths and weaknesses, they also answer different questions about metabolism. EM analysis is applied to study well-known, often manually compiled, small metabolic networks (less than hundred reactions) in order to gain insight into the physiology of particular organisms (Poolman, et al., 2003; Van Dien and Lidstrom, 2002) or to find elementary modes that have high molar yields for a compound of interest (Carlson, et al., 2004; Liao, et al., 1996). Knowledge of such elementary modes helps to produce desired compounds more efficiently (Trinh, et al., 2009), e.g. by modifying the compound-producing organism (Trinh, et al., 2006).

Path finding tools are obviously not suited for this kind of detailed analysis. In contrast to EM analysis, they are applied to large metabolic networks (up to several thousands of reactions), with the purpose of predicting biosynthesis and/or biodegradation pathways in organisms or sets of organisms (such as microbial communities) (e.g. (Dimitrov, et al., 2004; Pazos, et al., 2005; Yamazaki, et al., 2004)) or to fill gaps in genome-wide metabolic pathway reconstruction (Kharchenko, et al., 2006; Kharchenko, et al., 2004). Future application of path finding in metabolic reconstruction may go beyond gap filling. Current automated metabolic reconstruction tools (e.g. (Karp, et al., 2002; Moriya, et al., 2007)) are based on the mapping of enzymes onto pre-defined pathways, which prevents the detection of variants of known pathways and novel pathways. The more flexible path finding approach may help to solve these problems. Currently, only MaGe (microbial genome annotation system (Vallenet, et al., 2006)) integrates a path finding tool (PHT). Another interesting application of path finding could be to predict pathways from gene expression data, more specifically from sets of activated/repressed enzymes (van Helden, et al., 2001).

To summarize: EM analysis is suitable for detailed analysis of small-scale, high-quality metabolic networks, whereas path finding is designed to predict pathways in large-scale, possibly low-quality metabolic networks.

In our opinion, EM and path finding should be perceived as complementary tools that can be combined to explore metabolic diversity. For example, path finding might be used as a first step to discover novel pathways from various data sources (operons, synteny, gene expression), which might then be submitted to EM analysis in order to extract consistent modules of reactions. More generally, each approach is designed to answer different questions, and cannot be compared to methods conceived for other purposes. For example, Panke and co-workers analyze the dynacmis of in vitro enzymatic systems using a formalism that relies on the precise measurement of each enzyme’s condition (Hold and Panke, 2009; Makart, et al., 2007). Palsson’s group applies flux balance analysis to model genome-scale metabolic networks, in order to predict mutant phenotypes under controlled conditions, and design mutants with improved biosynthetic efficiency (Edwards and Palsson, 2000; Herrgård, et al., 2006; Portnoy, et al., 2008). Yet other methods have been developed to address different metabolism-related problems, whose enumeration is out of scope for this comment.

Rather than comparing EM-based tools to path finding tools, it would be much more interesting to compare path finding tools among themselves, or, even better, to evaluate the best ways to combine these tools. Indeed, many path finding approaches have been published that address the same question, but rely on different strategies (compound similarity, node or edge weights, rules) (Arita, 2003; Blum and Kohlbacher, 2008; Croes, et al., 2005; Croes, et al., 2006; Ellis, et al., 2008; Faust, et al., 2009; McShan, et al., 2003; Pazos, et al., 2005; Rahman, et al., 2004).

Currently, no approach is able to predict biochemically valid pathways in a genome-scale metabolic network with 100% accuracy, leaving a challenge for future research. Yet a greater challenge will be to perform experimental validations of the predictions, which, so far, has been done only in a very restricted number of cases (Table 1).

ACKNOWLEDGEMENTS

Funding: KF is supported by Actions de Recherches Concertées de la Communauté Française de Belgique (ARC grant number 04/09-307). The laboratory is a member of the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265). The work was also supported by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet).

We would like to thank the referees for their helpful comments.

REFERENCES

Arita, M. (2003) In Silico Atomic Tracing by Substrate-Product Relationships in Escherichia coli Intermediary Metabolism, *Genome Research*, **13**, 2455-2466.

Berg, J.M., Tymoczko, J.L. and Stryer, L. (2002) *Biochemistry*. W.H. Freeman and Company, New York.

Blum, T. and Kohlbacher, O. (2008) MetaRoute: fast search for relevant metabolic routes for interactive network navigation and visualization *Bioinformatics*, **24**, 2108-2109.

Carlson, R., Wlaschin, A. and Srienc, F. (2004) Kinetic Studies and Biochemical Pathway Analysis of Anaerobic Poly-(R)-3-Hydroxybutyric Acid Synthesis in Escherichia coli *Applied and Environmental Microbiology*, **71**, 713-720.

Croes, D., Couche, F., Wodak, S. and van Helden, J. (2005) Metabolic PathFinding: inferring relevant pathways in biochemical networks, *Nucleic Acids Res*, **33**, W326-W330.

Croes, D., Couche, F., Wodak, S. and van Helden, J. (2006) Inferring Meaningful Pathways in Weighted Metabolic Networks, *J. Mol. Biol.*, **356**, 222-236.

de Figueiredo, L.F., Schuster, S., Kaleta, C. and Fell, D.A. (2008) Can sugars be produced from fatty acids? A test case for pathway analysis tools *Bioinformatics*, **24**, 2615-2621.

Dimitrov, S., Kamenska, V., Walker, J.D., Windle, W., Purdy, R., Lewis, M. and Mekenyan, O. (2004) Predicting the biodegradation products of perfluorinated chemicals using CATABOL, *SAK and QSAR in Environmental Research*, **15**, 69-82.

Edwards, J.S. and Palsson, B.O. (2000) Metabolic flux balance analysis and the in silico analysis of Escherichia coli K-12 gene deletions, *BMC Bioinformatics*, **1:1**.

Ellis, L., Gao, J., Fenner, K. and Wackett, L. (2008) The University of Minnesota pathway prediction system: predicting metabolic logic, *Nucleic Acids Res*, **36**, W427-W432.

Faust, K., Croes, D. and van Helden, J. (2009) Metabolic pathfinding using RPAIR annotation., *J. Mol. Biol.*, **388**, 390-414.

Herrgård, M.J., Fong, S.S. and Palsson, B.Ø. (2006) Identification of Genome-Scale Metabolic Network Models Using Experimentally Measured Flux Profiles **2:e72**.

Hold, C. and Panke, S. (2009) Towards the engineering of in vitro systems, *Journal of the Royal Society Interface*, **6**, S507-S521.

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T. and Yamanishi (2008) KEGG for linking genomes to life and the environment, *Nucleic Acids Research*, **36**, D480-484.

Karp, P.D., Paley, S. and Romero, P. (2002) The Pathway Tools software, *Bioinformatics*, **18**, S225-S232.

Kharchenko, P., Chen, L., Freund, Y., Vitkup, D. and Church, G.M. (2006) Identifying metabolic enzymes with multiple types of association evidence *BMC Bioinformatics*, **7:177**.

Kharchenko, P., Vitkup, D. and Church, G.M. (2004) Filling gaps in a metabolic network using expression information *Bioinformatics*, **20**, i178-i185.

Liao, J.C., Hou, S.-Y. and Chao, Y.-P. (1996) Pathway Analysis, Engineering, and Physiological Considerations for Redirecting Central Metabolism *Biotechnology and Bioengineering*, **52**, 129-140.

Makart, S., Bechtold, M. and Panke, S. (2007) Towards preparative asymmetric synthesis of β-hydroxy-α-amino acids: L-allo-Threonine formation from glycine and acetaldehyde using recombinant GlyA, *Journal of Biotechnology*, **130**, 402-410.

McShan, D.C., Rao, S. and Shah, I. (2003) PathMiner: predicting metabolic pathways by heuristic search, *Bioinformatics*, **19**, 1692-1698.

Moriya, Y., Itoh, M., Okuda, S., Yoshizawa, A.C. and Kanehisa, M. (2007) KAAS: anautomatic genome annotation and pathway reconstruction server, *Nucleic Acids Res*, **35**, W182–W185.

Moult, J., Fidelis, K., Kryshtafovych, A., Rost, B., Hubbard, T. and Tramontano, A. (2007) Critical assessment of methods of protein structure prediction—Round VII *Proteins*, **69**, 3-9.

Pazos, F., Guijas, D., Valencia, A. and De Lorenzo, V. (2005) MetaRouter: bioinformatics for bioremediation, *Nucleic Acids Res*, **33**, D588-D592.

Pfeiffer, T., Sanchez-Valdenebro, I., Nuno, J.C., Montero, F. and Schuster, S. (1999) METATOOL: for studying metabolic networks., *Bioinformatics*, **15**, 251-257.

Planes, F.J. and Beasley, J.E. (2008) A critical examination of stoichiometric and path-finding approaches to metabolic pathways, *Brief Bioinformatics*, **9**, 422-436.

Poolman, M.G., Fell, D.A. and Raines, C.A. (2003) Elementary modes analysis of photosynthate metabolism in the chloroplast stroma *Eur. J. Biochem.*, **270**, 430-439.

Portnoy, V.A., Herrgård, M.J. and Palsson, B.Ø. (2008) Aerobic Fermentation of D-Glucose by an Evolved Cytochrome Oxidase-Deficient Escherichia coli Strain, *Applied and Environmental Microbiology*, **74**, 7561-7569.

Rahman, S.A., Advani, P., Schunk, R., Schrader, R. and Schomburg, D. (2004) Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC), *Bioinformatics*, **21**, 1189-1193.

Remde, A. and Debus, R. (1996) BIODEGRADABILITY OF FLUORINATED SURFACTANTS UNDER AEROBIC AND ANAEROBIC CONDITIONS *Chemosphere*, **32**, 1563-1574.

Schwartz, J.M. and Kanehisa, M. (2006) Quantitative elementary mode analysis of metabolic pathways: the example of yeast glycolysis *BMC Bioinformatics*, **7:186**.

Teusink, B., Passarge, J., Reijenga, C.A., Esgalhado, E., van der Weijden, C.C., Schepper, M., Walsh, M.C., Bakker, B.M., van Dam, K., Westerhoff, H.V. and Snoep, J.L. (2000) Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry *Eur. J. Biochem.*, **267**, 5313-5329.

Trinh, C.T., Carlson, R., Wlaschin, A. and Srienc, F. (2006) Design, construction and performance of the most efficient biomass producing E. coli bacterium *METABOLIC ENGINEERING*, **8**, 628-638.

Trinh, C.T., Wlaschin, A. and Srienc, F. (2009) Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism, *Appl Microbiol Biotechnol* **81**, 813-826.

Vallenet, D., Labarre, L., Rouy, Z., Barbe, V., Bocs, S., Cruveiller, S., Lajus, A., Pascal, G., Scarpelli, C. and Medigue, C. (2006) MaGe: a microbial genome annotation system supported by synteny results *Nucleic Acids Research*, **34**, 53-65.

Van Dien, S.J. and Lidstrom, M.E. (2002) Stoichiometric Model for Evaluating the Metabolic Capabilities of the Facultative Methylotroph Methylobacterium extorquens AM1, with Application to Reconstruction of C3 and C4 Metabolism, *Biotechnology and Bioengineering*, **78**, 296-312.

van Helden, J., Gilbert, D., Wernisch, L., Schroeder, M. and Wodak, S. (2001) Application of Regulatory Sequence Analysis and Metabolic Network Analysis to the Interpretation of Gene Expression Data *Lecture Notes in Computer Science*, **2066**, 147-165.

van Helden, J., Wernisch, L., Gilbert, D. and Wodak, S. (2002) Graph-based analysis of metabolic networks. In Mewes, H.-W., Weiss, B. and Seidel, H. (eds), *Ernst Schering Res. Found. Workshop*. Springer-Verlag, Heidelberg, 245-274.

von Kamp, A. and Schuster, S. (2006) Metatool 5.0: fast and flexible elementary modes analysis, *Bioinformatics*, **22**, 1930-1931.

Yamazaki, Y., Kitajima, M., Arita, M., Takayama, H., Sudo, H., Yamazaki, M., Aimi, N. and Saito, K. (2004) Biosynthesis of Camptothecin. In Silico and in Vivo Tracer Study from [1-13C]Glucose1 *Plant Physiology*, **134**, 161-170.

Table 1. Selected examples of application cases of metabolic predictions that have been confirmed experimentally.

Reference	Question	Approach	Tool	Network size	Experimental confirmation method
Yamazaki et al. (2004)	Enumeration of possible pathways to produce a compound of interest (Camptothecin).	Path finding (k-shortest paths)	ARM (Atomic Reconstruction of Metabolism)	131 reactions, 280 atom mappings	^{13}C tracing in vivo
Trinh et al. (2008)	Design of mutant *Escherichia coli* strains to optimize biomass production	EM	METATOOL	44 reactions, 47 metabolites	Gene knock-outs in combination with biomass yield measurements
Carlson et al. (2005)	Enumeration of possible pathways to produce a compound of interest (poly- -hydroxybutyrate (PHB)).	EM	METATOOL	44 reactions 47 metabolites	Measurement of PHB, glucose and other compound concentrations at different time points
Dimitrov et al. (2004)	Prediction of compound biodegradability (perfluorinated chemicals).	Rule-based pathway prediction	CATABOL	24 transformations derived from > 150 compounds participating in biodegradation pathways	Biodegradability tests (Remde and Debus, 1996)

6 Contributions to NeAT

Presented articles:
S. Brohée, K. Faust, G. Lima-Mendez, O. Sand, R. Janky, G. Vanderstocken, Y. Deville and J. van Helden
NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways
Nucleic Acids Research, vol. 36, pp. W444-W451, 2008.

K. Faust[1], S. Brohee, Gipsi Lima-Mendez, G. Vanderstocken and J. van Helden
Network Analysis Tools: from biological networks to clusters and pathways
Nature Protocols, vol. 3, pp. 1616-1629, 2008.

6.1 Presentation of NeAT

NeAT is a publicly available set of tools dedicated to the analysis of biological networks [20, 52]. Most NeAT tools can handle large input networks and finish jobs within one minute. NeAT can be accessed via a web server (http://rsat.ulb.ac.be/neat/), a collection of web services (SOAP/WSDL interface) as well as on command line. The web server, web services and the command line version are distributed on request and can be installed locally. NeAT is designed to be modular, to allow for easy combination of tools and implementation of workflows. Tools can be combined programmatically via command line pipes and web services or via mouse clicks on the web server. Each web server tool provides a number of buttons that appear after completion of a job and which allow to send the results as input to another NeAT tool without having to copy-paste. In addition, NeAT is documented on several levels: Each tool comes with its command line and WSDL documentation. On the web server level, a manual, a tutorial and at least one demo accompany most tools.

6.2 Tools contributed to NeAT

Table 6.1 lists the tools that I contributed to NeAT during my thesis. The tools make use of a database containing metabolic data from KEGG and BioCyc. This database is described in section 9.3.

[1] K. Faust and S. Brohée contributed equally to this publication.

Table 6.1: Tools contributed to NeAT

Tool name	Purpose	Data sources	Input	Output	Options	Implementation	Remarks
Pathfinder	Enumeration of k-shortest paths in weighted networks	None	Network, start node set, end node set.	Paths table or network combining paths	Input network format (gml [72] or tab-delimited), maximal path weight, maximal path length, minimal path length, path rank index, nodes to be absent from paths, nodes to be present in paths, weight policy (degree, unit or custom), output type (table or network), output network format (gml or tab), results sent by email or displayed in browser. Advanced options: Flag network as metabolic (will affect weight policy and reaction directionality treatment), exclusion attribute, alternative path finding algorithm [31, 32].	Java wrapper calling REA routine [83] and using the aMAZE graph library. Web interface written in PHP.	Sets of start and end nodes are supported using a graph transformation described in [47].
Metabolic pathfinder	Enumeration of metabolic paths in weighted KEGG networks	KEGG LIGAND, KEGG RPAIR	EC numbers, reactions, reactant pairs, compounds	Paths table linked to KEGG entries or network combining paths, with network image.	Network (KEGG LIGAND, KEGG RPAIR or a mixture of both), weight policy, path rank index, minimal path length, maximal path length, maximal path weight, nodes to be absent from paths, nodes to be present in paths, output type (table or network), output network format (gml, VisML [76], dot [44] or tab), results sent by email or displayed in browser. Advanced options: custom KEGG network (in tab, gml or KGML format), custom weights.	Set of Java servlets and jsp pages. Calls the Pathfinder web service and queries the metabolic database using the Hibernate query language (HQL, [71]) to annotate paths.	This tool does not exist on command line or as a web service, since it is a client of the Pathfinder web service.
KEGG network provider	Construction of networks specific to a set of organisms.	KEGG PATHWAY	KEGG organisms, KEGG reactions	Network	Network filtered for compounds, reactions or RPAIR classes. Construction of RPAIR networks, directed networks and networks respecting irreversible reactions. Node attributes (EC numbers, reaction equations, compound labels and compound formulas), Network format (tab, gml, dot or VisML). Results sent by email or displayed in browser.	Implemented in Java, metabolic database is queried with HQL to set node attributes. Networks are constructed from locally stored KGML files and saved, so that they need to be constructed only once. Web interface is written in PHP.	Conversion to KEGG RPAIR is offered, which is an option not present in other KEGG PATHWAY parsers [145, 174, 116].

Table 6.1: Tools contributed to NeAT

Tool name	Purpose	Data sources	Input	Output	Options	Implementation	Remarks
Network converter	Conversion of networks into another format	None	Network in tab-delimited, gml, GDL, KGML or biopax [16] format	Network in tab-delimited, gml, dot, Pajek [10], VisML or GDL format.	None	Implemented in Java. GDL is an XML-based graph format developed by the aMAZE team.	This tool has no web interface (because NeAT tool convert-graph fulfills a similar purpose), but it is available as a web service and on command line. It is used within NeAT to convert graphs into VisML format.
Pathway extraction	Extraction of a sub-network from a weighted network given a set or sets of seed nodes.	KEGG LIGAND, KEGG RPAIR, MetaCyc	Network, seed nodes. For pre-loaded networks, seeds can be gene names, UniProt identifiers, RefSeq identifiers, enzyme names, compounds, reactions, EC numbers or reactant pairs. Seeds can be grouped in sets.	Sub-network (file and image) with a table listing some of its topological properties. For pre-loaded networks, reference pathways are mapped onto the sub-network.	Pre-loaded network (MetaCyc, KEGG RPAIR, KEGG LIGAND) or custom network (in tab, gml or KGML format), weight policy or custom weights, extraction algorithm (repetitive REA, Takahashi-Matsuyama [154], kWalks [48, 21] or combinations thereof), preprocess, postprocess, iteration number, percentage of input network to be extracted in a hybrid approach, computation of weights with kWalks, sub-network format (tab-delimited, VisML, gml, dot), seed group treatment.	Implemented in Java with calls to external routines written in C for REA and kWalks. The metabolic database is queried via HQL to annotate metabolic sub-networks and to map reference pathways. Web interface is implemented with JSP and Java servlets.	

6.3 On the use of NeAT tools

6.3.1 Data input

Users can up-load their data to NeAT in two main formats: as a network (graph) or as a tab-delimited table (note that tables can be interpreted as networks as well). NeAT supports a variety of graph formats: tab-delimited (table with two selected columns), DOT [44], GML [72] and some specialized tools also KGML, the XML format of KEGG. In addition, users can retrieve networks from the two database interfaces offered by NeAT: one for STRING [81] and the second for KEGG PATHWAY, which store data on protein-protein interactions and metabolism respectively.

Usually, network nodes represent biological objects whose interactions have been determined by a variety of methods. For instance, a user interested in the analysis of protein interactions might up-load a protein-protein interaction network having proteins as nodes and their interactions as edges. In practice, this could be a tab-delimited file with two protein name columns, where each row describes the interaction between two proteins. The user could then up-load a second protein-protein interaction network and measure the overlap between the two networks (see details in the Nature Protocol attached to this chapter) or compare the network with experimentally obtained protein-protein interactions stored in STRING (experimental data channel). With the path finder tool, the user could in addition enumerate paths between a pair of proteins of interest.

6.3.2 Data output and interpretation

The main outputs of NeAT are tab-delimited tables displaying statistics or networks. Networks can be returned in several formats: GML, DOT, VisML [76], tab-delimited and as adjacency matrix.

For the path finding and pathway extraction tools in NeAT, it is often not easy to interpret the output. For instance, the pathfinder tool lists paths in the order of their length (unweighted networks) or their weight (weighted networks). In order to filter out irrelevant paths, users have several possibilities: First and foremost, they can assign network weights that favor certain nodes and penalize others. In addition, they can indicate nodes that have to appear in paths or that should be avoided. Moreover, they can restrict the path length. For example, if a user would like to predict a signal transduction pathway from a protein-protein interaction network and has an idea about the minimal length (say 8 steps) of this pathway and some of its participating proteins (e.g. STE20 and STE7), he or she can set these constraints in the web form as *minimal length* and *nodes to be included* parameter values.

The pathway extraction tool outputs only one solution, namely the predicted pathway. However, the interpretation of a predicted pathway, especially if large, is not straightforward. Criteria that could be taken into account for interpretation are:

- *Seed node coverage.* A high percentage of seed nodes among all pathway nodes indicates that the predicted pathway is strongly supported by the seed nodes.

- *Distance between seed nodes.* The reliability of the predicted pathway decreases with the distance between seeds. For instance, a long branch not supported by any seed node except the terminal one very likely represents a false positive.

- *Presence of enyzmes catalyzing predicted reactions in the query organism.* The predicted pathway consists of seed nodes and predicted nodes connecting the seeds. A pathway is more reliable if all its predicted reactions are likely to be catalyzed in the organism of interest. However, reactions not associated to any gene in the query organism might also be spontaneous. Also, some enzymes may be missing due to errors in the genome annotation process.

- *Presence of predicted nodes in the source data set.* Closer inspection of the source data set may indicate that seed nodes and some predicted nodes belong to the same group. For instance, in a microarray data set, genes may have been removed from a cluster because their expression ratios were below the given threshold. However, these genes may be part of the pathway connecting the seed genes.

- *Reproducibility.* If several data sources exist for the same kind of data, pathway prediction can be repeated in a network constructed from an alternative source. For example, the pathway extraction tool offers metabolic networks from databases assembled by two independent groups, namely the KEGG and MetaCyc teams. The reproduction of a pathway discovered in the KEGG network with the MetaCyc network may increase the confidence in this pathway.

- *Known pathways.* Known pathways overlapping with the extracted pathway may help to interpret the latter one. For instance, the extracted pathway may be a combination of known pathways or may combine a known pathway with a novel one. To ease interpretation, the pathway extraction tool displays links to the overlapping pathways annotated in KEGG or MetaCyc (depending on the network used) and computes the significance of their overlap using various metrics (e.g. Jaccard similarity).

W444–W451 *Nucleic Acids Research, 2008, Vol. 36, Web Server issue* *Published online 4 June 2008*
doi:10.1093/nar/gkn336

NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways

Sylvain Brohée[1,*], Karoline Faust[1], Gipsi Lima-Mendez[1], Olivier Sand[1], Rekin's Janky[1], Gilles Vanderstocken[1], Yves Deville[2] and Jacques van Helden[1]

[1]Laboratoire de Bioinformatique des Génomes et Réseaux (BiGRE), Université Libre de Bruxelles (ULB), Boulevard du Triomphe, CP263, B-1050 Bruxelles and [2]Department of Computing Science and Engineering, Université catholique de Louvain (UCL), Place Sainte Barbe, 2. B-1348 Louvain-la-Neuve, Belgium

Received January 31, 2008; Revised April 24, 2008; Accepted May 10, 2008

ABSTRACT

The network analysis tools (NeAT) (http://rsat.ulb.ac.be/neat/) provide a user-friendly web access to a collection of modular tools for the analysis of networks (graphs) and clusters (e.g. microarray clusters, functional classes, etc.). A first set of tools supports basic operations on graphs (comparison between two graphs, neighborhood of a set of input nodes, path finding and graph randomization). Another set of programs makes the connection between networks and clusters (graph-based clustering, cliques discovery and mapping of clusters onto a network). The toolbox also includes programs for detecting significant intersections between clusters/classes (e.g. clusters of co-expression versus functional classes of genes). NeAT are designed to cope with large datasets and provide a flexible toolbox for analyzing biological networks stored in various databases (protein interactions, regulation and metabolism) or obtained from high-throughput experiments (two-hybrid, mass-spectrometry and microarrays). The web interface interconnects the programs in predefined analysis flows, enabling to address a series of questions about networks of interest. Each tool can also be used separately by entering custom data for a specific analysis. NeAT can also be used as web services (SOAP/WSDL interface), in order to design programmatic workflows and integrate them with other available resources.

INTRODUCTION

During the last decade, large-scale biological studies produced huge amounts of data that reveal various layers of molecular interaction networks: protein interactions, transcriptional regulation, metabolic reactions, signal transduction, etc.

Graphs (in the mathematical sense) have been used to represent, study and integrate such biological networks. By definition, a mathematical graph is a set of nodes (generally represented as dots) that are connected by edges (lines between dots). Edges may be enriched by several features, e.g. a direction (an edge from node *A* to node *B* is distinct from an edge from *B* to *A*), a color, a type and a weight (a value is associated to the edges).

Such edges and nodes provide convenient ways to represent biological features. For example, in a protein–protein interaction network, a node represents a polypeptide and an edge indicates the existence of a physical interaction between two polypeptides (1). A weight can optionally be put on edges to reflect the strength of interactions. In 'compound-centric' metabolic networks, nodes represent metabolites and the directed edges represent the enzymes used to convert a metabolite into another one (2). The metabolic networks may also be represented as bipartite graphs, i.e. a network with two distinct types of nodes (one for compounds and one for reactions), where edges must always link a node of one type to a node of the other type (3,4). Similarly, graphs can be used to represent regulatory relationships (5,6) and transduction pathways (7). Network biology is emerging as a very fertile field, as reflected by the rapidly increasing pace of relevant publications (8,9).

Despite the ever-increasing availability of data that may be represented as networks, large-scale analyses should be considered with caution, for several reasons. Firstly, high-throughput data are notoriously noisy (presence of false positives) and incomplete (10,11). In addition, some interaction networks have been characterized by several independent studies, which are providing complementary subsets of the data. Important efforts will thus be required to extract reliable information from the ever-increasing amount of data.

Specialized tools are required to extract and compare information obtained from multiple data sources, and

*To whom correspondence should be addressed. Tel: +32 02 6505434; Fax: +32 02 6505425; Email: sylvain@scmbb.ulb.ac.be

apply various statistical parameters treatments to describe and understand networks properties. For this purpose, we developed the *Network Analysis Tools* (*NeAT*), a set of modular software tools supporting a large variety of operations on networks and clusters. The web interface provides a convenient and intuitive access to the tools and allows to thread user-provided data sets through typical analysis work flows, in order to interpret their networks. The NeAT programs may be grouped in three categories: tools for manipulating graphs (graph comparison, randomization, alteration, visualization, etc.), tools for analyzing clusters (or, equivalently, classes) (cluster comparison, etc.) and tools that establish the link between networks and clusters (graph clustering, graph–cluster mapping, etc.).

NeAT DESCRIPTION

Figure 1 and Table 1 present the collection of tools available in NeAT as well as their input and output types. On the website, each tool is accessible via the menu on the left panel of the web page (Figure 3, inset).

As shown in Table 1, NeAT tools can be broadly grouped in three categories: *network tools* perform various operations on one or several graphs, *cluster tools* are mainly dedicated to comparisons between clusters and *network–clusters* tools make the connection between networks and clusters.

We will briefly describe the function of each tool together and discuss some typical application. Further information and examples of utilization can be found in the cited literature.

NETWORK TOOLS

Network topology

Several statistics have been defined to characterize global topological properties of a network. It has been shown that these topological properties distinguish biological networks from random networks. Noticeably, it is often stated that the distribution of degree (the number of edges connected per nodes) follows a power-law distribution (12). The program *graph-topology* computes the degree of each node of a graph, which can then be analyzed either as a full result table or visualized as a *XY* plot (Figure 2). *Graph-topology* also computes the betweenness (i.e. the proportion of shortest going through a node) and the

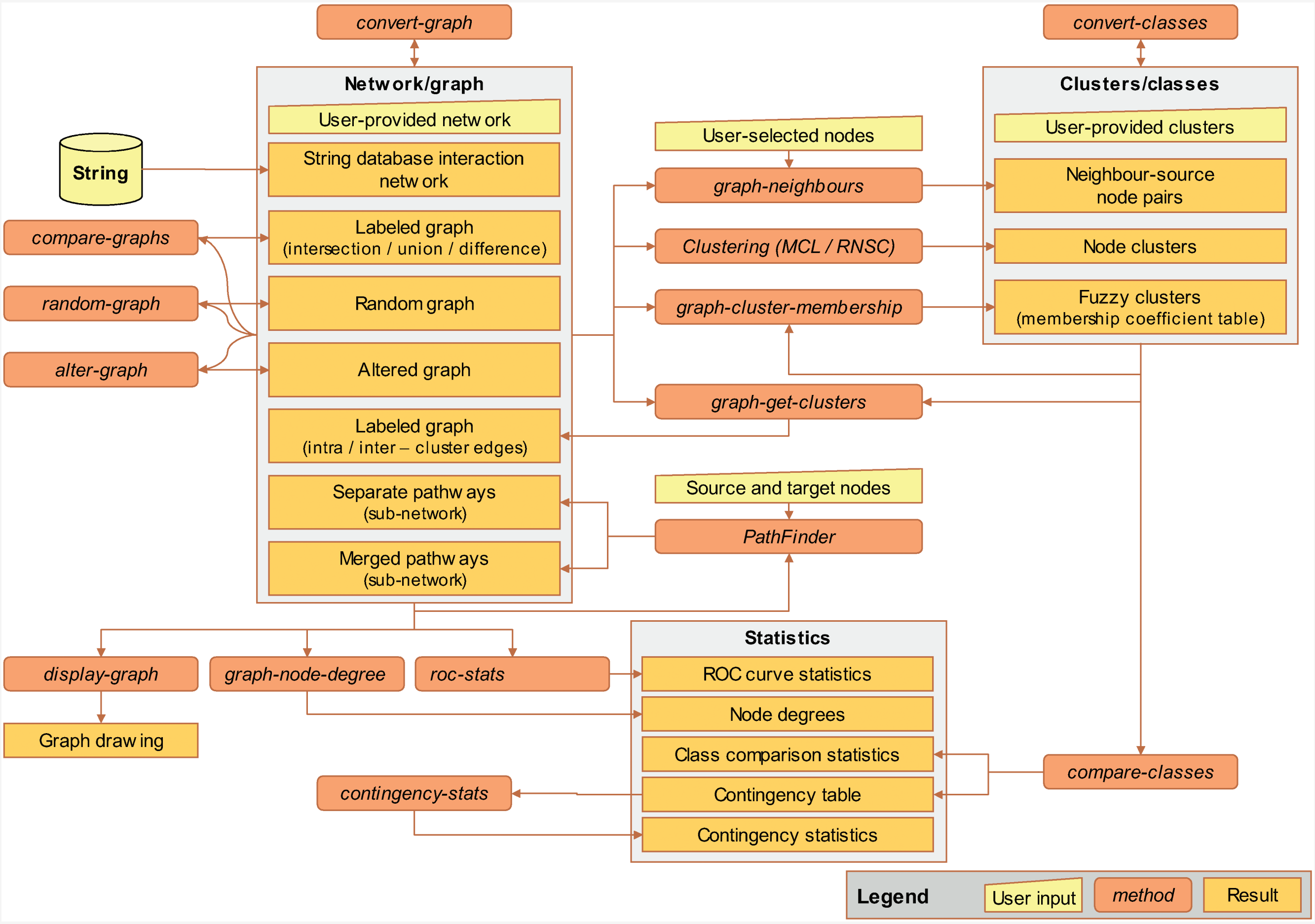


Figure 1. Flow chart of the tools and data types supported on NeAT. Trapezoidal boxes represent user-provided input, rounded boxes programs and rectangles results.

Table 1. Description of the programs available in NeAT

Program	Description	Input	Output
Network tools			
convert-graph	Converts a graph from a format to another one, position the nodes and changes the edge colors and width according to its weight	A network in a given format	A network in the requested format
display-graph	Draws a network graphical representation	A network	A figure in the requested format
compare-graphs	Computes the intersection, the union or the difference of two networks	Two networks to be compared	A network (intersection, union, difference)
random-graph	Generates random graphs either from an existing graph or from scratch according to different randomization procedure.	A graph or a list of node names or nothing	A randomized network
graph-topology	Calculates the degree, betweenness and closeness of each node and specifies if this node is a source or a target node	A network, (list of nodes for which the degree has to be computed)	A table the requested centrality statistics of each requested node
alter-graph	Alters a graph either by adding or removing edges or nodes (targeted removal or not)	A network	An altered network
Pathfinder	Finds the *k*-shortest path between a set of source nodes and a set of target nodes	A network and the list of source and target nodes	A table of pathway or a network composed of the set of pathways
String dataset download	Downloads a subset of the network of the String database (40)	A list of nodes for which you want to know the neighbors in String	The neighbors of the nodes your entered in and the edges between them.
Network clusters tools			
MCL (34,36) *and RNSC* (37)	Finds the densely connected subsets of the graph	A network	A list of clusters
graph-cligue	Extract al cliques from a graph	A network	A list of cliques
graph-neighbours	Extracts from a graph the neighborhood of a set of seed nodes	A network, (a list of seed nodes)	Clusters of neighbor-source node pairs
graph-cluster-membership	Maps a clustering result onto a graph and compute the membership degree between each node and each cluster, on the basis of edges linking this node to the cluster	A network, clustering results	A tab-delimited membership table, where each row represents a node and each column a cluster. Entries are the membership degree of the node.
graph-get-clusters	Compares a graphs with clusters. Extracts the intra-clusters edges or map the clusters on the network	A network	An edge-labeled network
Clusters tools			
compare-classes	Compares two class files (the query file and the reference file). Each class of the query file is compared to each class of the reference file.	One or two cluster files	For each comparison, the number of common elements and comparison statistics or a contingency table
contingency-stats	Study of a contingency table	A contingency table	Statistics according to ref. (26)
Others tools			
roc-stats	Calculates and draws ROC curve	Scored results associated with validation labels	For each score value, the derived statistics (Sn, PPV, FPR), which can be further used to draw a ROC curve.

Input Parameters between brackets are optional.

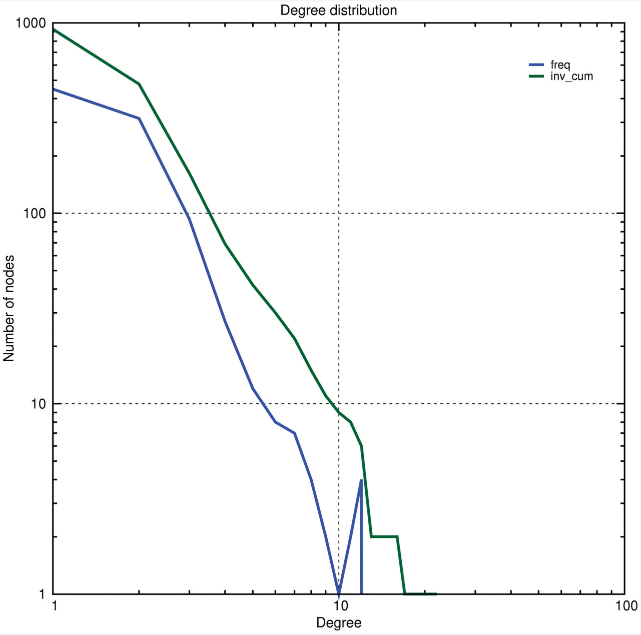


Figure 2. Node degree distribution of a yeast protein interaction network obtained from two-hybrid data. The distribution was computed with the program *graph-topology* and plotted on log scales for both the abscissa and ordinates. The linear shape of the curve on the log–log graph suggests that this network follows a power-law distribution of degree. Color code : blue, absolute frequency; green, reverse cumulative frequency.

closeness (i.e. the mean shortest distance of a node to all others) of each node in the network.

Node neighborhood

Starting from one or several nodes of interest, the program *graph-neighbours* collects neighbor nodes up to a user-specified distance. Neighborhood analysis can be for example applied to predict the function of an unknown polypeptide by collecting its neighbors with known function in a protein interaction network ('guilty by association') (13).

Network comparison

The program *compare-graphs* computes the intersection, the union and/or the difference between two input networks and estimates the statistical significance of the overlap (Figure 3, inset).

These basic operations between graphs can serve for many other tasks: the union can be used to integrate networks at different layers (e.g. metabolism, transduction signal and transcriptional regulation), the intersection to select interactions with evidences in two distinct experiments, the differences to select interactions detected by one method and missed by another one. A typical example of application is to estimate the relevance of a protein–protein interaction network obtained by some high-throughput experiment, by comparing it with a manually curated network [e.g. BioGrid or MIPS databases data (14,15)].

Evaluation of predicted networks using receiver operating characteristic (ROC) curves

The program *roc-stats* is typically used as a postanalysis program after a network comparison between predicted and annotated networks. It takes as input a set of scored results associated with validation status (positives or negatives) and computes, for each threshold on the score, the derived statistics: true positive rate (TPR, also called sensitivity), positive predictive value (PPV), false positive rate (FPR) and accuracy.

Those statistics are also further used to draw different graphical plots showing the performance as a function of the score threshold or allowing performance comparisons (precision recall and ROC curves).

ROC curves show the fraction of the true positives (TPR) versus the fraction of the (FPR) and are often used to compare the predictive performance of different programs (16).

Path finding in a network

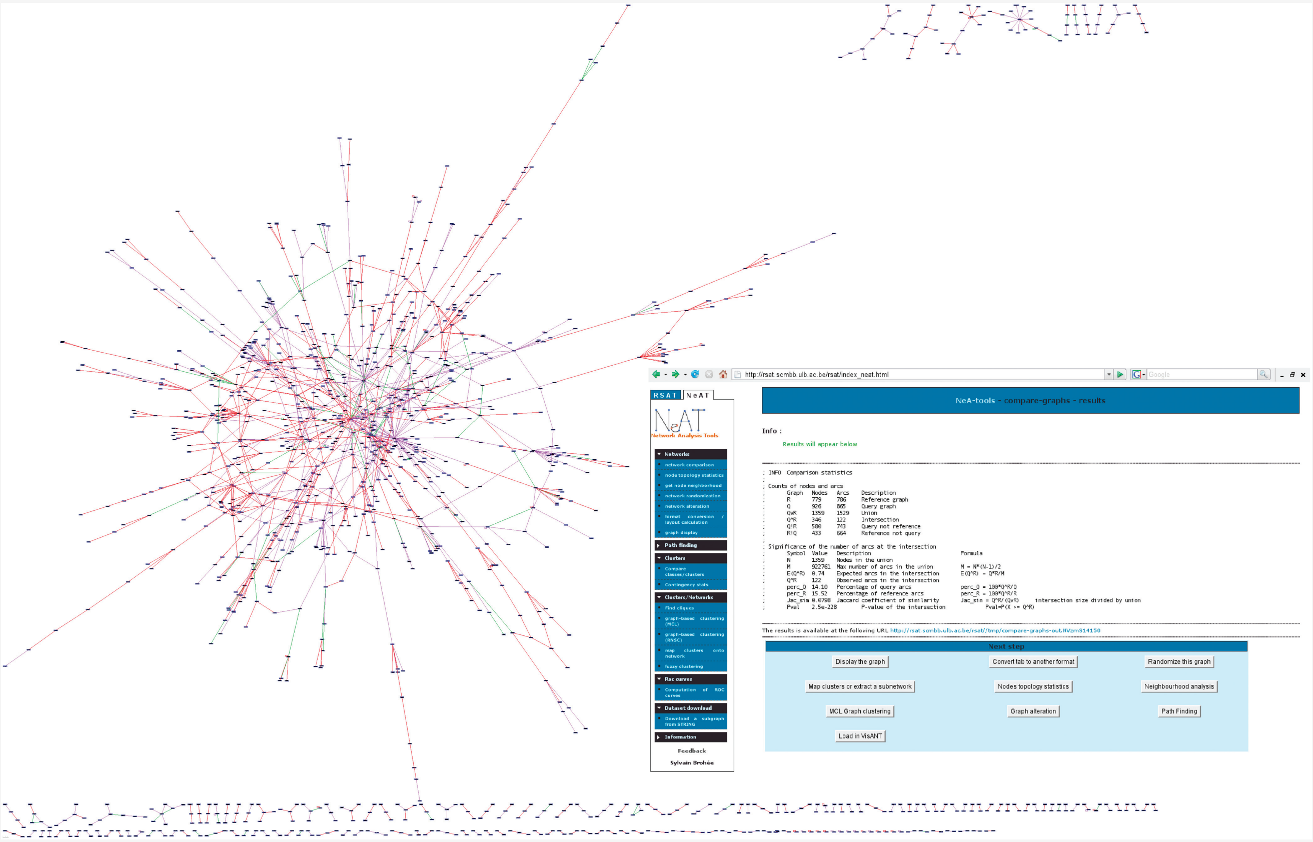
Biochemical interactions form intricate networks, where a multitude of pathways can be used to join two nodes of interest. The search of optimal paths (minimizing the number of steps, or the distance, or some weight) has a long tradition in graph theory. Path finding algorithms have been applied to uncover signal transduction pathways from protein–protein interaction networks (17–19) or metabolic pathways in metabolic networks, respectively (20–22). Recently, we evaluated the performance of a *k*-shortest path finding algorithm for metabolic pathway inference and found that the correspondence between inferred and annotated pathways can be crucially improved by setting an appropriate weighting on the nodes of the metabolic network, in order to penalize highly connected compounds (4).

The NeAT interface includes a general *k*-shortest path finding algorithm, that supports searches from a set of (one or several) source nodes to a set of target nodes (23). Node weights can either be specified in the input graph or computed automatically according to node degree (4).

Network randomization and alteration

Random graphs are extremely useful to analyze statistical properties of graphs and to validate theoretical models (24–27). The significance of some properties observed in a biological network (e.g. node degree, clustering coefficient, network diameter, etc.) can be estimated by measuring the distribution of probability of the same property in a large set of random networks. Random networks can also be used to observe the behavior of a given algorithm (e.g. clustering) in absence of biological information.

The program *random-graph* supports different procedures to randomize a network, which can then be submitted to the same workflows in the same way as a real biological network. Random graphs can be generated from the scratch, according to an Erdös-Renyi model. Alternatively, random graphs can be generated by permuting the edges between the nodes of a given input graph. This randomization preserves the degree of each



Info :

Results will appear below

```
; INFO  Comparison statistics
;
; Counts of nodes and arcs
;       Graph   Nodes   Arcs    Description
;       R       779     786     Reference graph
;       Q       926     865     Query graph
;       QvR     1359    1529    Union
;       Q^R     346     122     Intersection
;       Q!R     580     743     Query not reference
;       R!Q     433     664     Reference not query
;
; Significance of the number of arcs at the intersection
;       Symbol  Value   Description                             Formula
;       N       1359    Nodes in the union
;       M       922761  Max number of arcs in the union         M = N*(N-1)/2
;       E(Q^R)  0.74    Expected arcs in the intersection       E(Q^R) = Q*R/M
;       Q^R     122     Observed arcs in the intersection
;       perc_Q  14.10   Percentage of query arcs                perc_Q = 100*Q^R/Q
;       perc_R  15.52   Percentage of reference arcs            perc_R = 100*Q^R/R
;       Jac_sim 0.0798  Jaccard coefficient of similarity       Jac_sim = Q^R/(QvR)     intersection size divided by union
;       Pval    2.5e-228        P-value of the intersection             Pval=P(X >= Q^R)
```

The results is available at the following URL http://rsat.scmbb.ulb.ac.be/rsat//tmp/compare-graphs-out.NVzmS14150

Figure 3. The *compare-graphs* result. Main figure: result of the comparison between two large-scale yeast protein interaction networks obtained by the two-hybrid method (41,42). The networks were compared using *compare-graphs* and displayed with yED. Edge color code: green, edges present in both networks (intersection); red, edges present in Ito's data set only; violet, edges present in Uetz' dataset only. Inset: comparison statistics, including an estimation of the significance of the intersection between the network comparison, based on the hypergeometric distribution.

node. A third mode of randomization preserves the degree distribution of the input graph, without preserving the degree of individual nodes.

Another tool, *alter-graph*, performs a partial randomization of a given input graph, by combining two operations: random addition and/or deletion of nodes and/or edges. Altered graphs are particularly useful to study the robustness of procedures to the presence of noise (node/edge additions) or to missing information (node/edge deletions). This tool was used in our comparative assessment of four graph-based clustering algorithms (26).

Network display

NeAT includes a tool called *display-graph*, which generates static images of an input network. Such drawings are convenient for a quick inspection of the results from the web browser, especially when dealing with large graphs. However, the cost of this speed is that the layout is rather rudimentary and the resulting image is static.

For more sophisticated layouts and for a dynamical manipulation of the drawing, NeAT is also able to load a network directly into the VisANT graph editor via Java Web Start (28).

For more advanced visualization facilities, we recommend specialized graph editors like yED Graph Editor (http://www.yworks.com/en/products_yed_about.html) and Cytoscape [(29), http://www.cytoscape.org]. To this purpose, the tool *convert-graph* permits to export any network resulting from NeAT to the GML format (http://www.infosun.fim.uni-passau.de/Graphlet/GML/gml-tr.html) which is supported by both editors.

CLUSTER TOOLS

NeAT also presents a series of tools allowing to study clusters or classification (functional classes) (Table 1, clusters tools). For example, the program *compare-classes* can study if among the clusters of highly connected nodes extracted from a graph via some clustering algorithm, some overlap with biological relevant classes [e.g. gene ontology classes (30)] exists. This program also allows to create a contingency table that can be further analyzed via the *contingency-stats* application.

NETWORK–CLUSTER TOOLS

Network clustering

Various algorithms have been implemented to extract clusters (i.e. groups of densely connected nodes) from biological networks. Clustering algorithms are often used

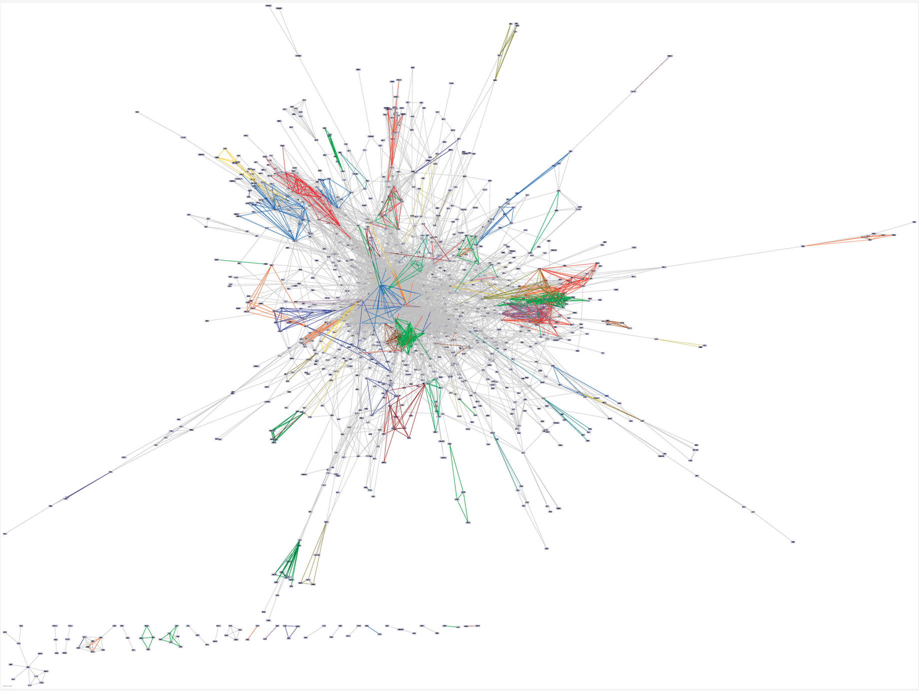


Figure 4. Comparison between a network and a set of classes. Mapping of the yeast protein complexes stored in MIPS database (15) on a large-scale interaction data set obtained by coimmunoprecipitation followed by mass spectrometry experiments (39). The mapping and coloring was performed with *graph-get-clusters*, and the image generated with the graphical editor *yED*. Intercluster edges (edges between nodes that do not belong to the same complex) are displayed in gray. Intracluster edges (edges between nodes belonging to the same complex) are colored with cluster-specific colors (one color for each protein complex).

in biology in order to extract coherent groups of nodes from networks : detection of protein complexes (26,31–33), of protein families (24), extraction of co-expressed clusters from in co-expression networks (35), etc.

The graph-based clustering algorithms MCL (34,36) and RNSC (37) have been shown to obtain good performances for extracting protein complexes from protein interaction networks (26). These algorithms can deal with large graphs and are very efficient in time. For these reasons, we included them in the NeAT tool suite.

Moreover, NeAT also includes a tool that discovers cliques (fully connected set of nodes) in networks.

From partitions to fuzzy clusters

As many clustering algorithms, MCL or RNSC partition the graphs into nonoverlapping clusters: each node is assigned to one and only one cluster. However, in some types of biological data, a single assignment may fail to represent multiple relationships between a node and various types of neighbors (for example, a protein may be part of different complexes).

Some graph-based clustering algorithms support multiple assignment and nonassigned nodes (i.e. fuzzy clustering), but the tuning of their parameters is sometimes delicate and the results can sometimes be weaker than those of a partitioning algorithm.

To keep the best of both worlds, an approach is to first run a partitioning algorithm and to postprocess its result by measuring *a posteriori* the membership between each node and each cluster of the partition. The membership of a node to a cluster is the proportion of edges from this node that reach that cluster. If the graph is weighted, the membership can take edge weights into account.

This two-step approach has been used to perform a reticulate classification of phages and detect mosaic phages resulting from fusions between other phage genomes (38). The program *graph-cluster-membership* takes as input a graph and a clustering result, and returns a node/cluster table indicating the degree of membership of each node to each cluster.

On the NeAT site, clustering results can automatically be launched to the *graph-cluster-membership* form. *graph-cluster-membership* can easily be adapted to be combined with other graph-based clustering algorithms.

Mapping of classes onto network

NeAT program *graph-get-clusters* then allows to extract or to map node clusters onto the network. A first function of such a mapping is to visualize the coherence of protein clusters or functional classes in the context of the network. Figure 4 displays a typical example of *graph-get-clusters* results, where known protein complexes (15) have been mapped onto a yeast protein interaction network obtained by high-throughput co-immunoprecipitation experiments (39). Edges between proteins belonging to annotated structural complexes have been colored according to their cluster (complex) membership. This helps the user to visualize the position of complexes in the interaction network.

DOCUMENTATION

NeAT programs are documented at various levels. Firstly, a manual is accessible from each query form, providing a systematic description of the parameters. Second, DEMO buttons automatically fill the query form with predefined examples (data sets + parameter values), in order to give the intuition of the result returned by the tools on a typical situation. Third, NeAT contains a tutorial, where users can learn using the tools on the basis of concrete biological data sets.

IMPLEMENTATION AND AVAILABILITY

Unless otherwise specified, all interaction data sets available in the NeAT demonstrations and the tutorials were downloaded from the BioGrid database (14) (http://www.thebiogrid.org/).

Moreover, NeAT includes a tool allowing to download and precisely filter subsets of the String database. This database contains protein interaction data obtained by integrating known and predicted interactions from a variety of sources (40).

Except for the path finding and the graph layout algorithms, all NeAT programs were developed in Perl and can be used as stand-alone applications on UNIX-based systems (tested on Linux + Mac OSX). The stand-alone version is freely available for academic users upon request (see *Informations* on the NeAT website).

The large majority of NeAT tools allows the treatment of graphs with several thousands of nodes and several tens of thousands of edges in a reasonable time. Typical published biological networks (a few thousands of nodes and tens of thousands of edges) are treated within seconds. However, some tools may be slower (cliques discovery (NP-hard), betweenness and closeness computation), but the execution time stays reasonable (minutes).

The web site (http://rsat.scmbb.ulb.ac.be/neat/) is free and open to all users and there is no login requirement.

NeAT programs are also accessible as web services (interface SOAP/WSDL), which allows to design programmatic workflows and integrate NeAT tools with various remote resources (databases and software tools). Actually, our website is itself a client for the web services, which guarantees a constant care for maintaining functional web services.

CONCLUSION

The Network Analysis Tools provide bioinformaticians and biologists with a set of web tools that can be combined to efficiently perform the main graph operations (comparison, node degree computation, clustering, etc.) used in today's network biology. As all programs can be integrated in workflows, either on our website or via SOAP web services, users can easily use them to study the topology of a network of interest, discover densely connected groups of nodes, compare these groups to some reference classification and run negative control by submitting randomized graphs to the same analysis.

With the increasing number of studies involving biological networks, we are confident that the NeAT web server will be useful to biologists in general and to network bioinformaticians in particular.

ACKNOWLEDGEMENTS

Pathfinder is a Java wrapper around the REA algorithm written by Jimenez and Marzal in C. K.F. acknowledge these developers for putting their code in public. The NeAT java wrapper was inspired by a Python wrapper written by Pierre Schaus and Jean-Noël Monette at UCL (INGI), who are acknowledged for having helped K.F. K.F. would also like to thank the former aMAZE team for their support and help. The clustering algorithm MCL was developed by Stijn Van Dongen (Sanger Institute, UK). We are thankful to Stijn for making his code publicly available, for allowing us to integrate it in our web server, and for his extensive feed-back on the NeAT web interface. The clustering algorithm RNSC was developed by Andrew King (McGill Institute, Canada). We are thankful to Andrew King for making his code publicly available and for allowing us to integrate it in our web server. We thank Zhenjun Hu for helping us to integrate VisANT in NeAT and Morgane Thomas-Chollier (Université Libre de Bruxelles, Belgium) for her help in the website design and in the implementation of php scripts. We also acknowledge the following colleagues for having tested NeAT and for their constructive comments: Cei Abreu (Sanger Institute, UK), Bruno Contreras Moreira (University of Saragossa, Spain), Alexandre Irrthum (Université de Liège, Belgium), Philippe Gautier (MRC Edimburg, UK), Nicolas Simonis (Harvard Medical School, Boston, USA) and Shuye Pu (Hospital for Sick Children, Toronto, Canada). S.B. and R.J. are recipients of a PhD grant from the Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture (FRIA). K.F. is supported by The Actions de Recherches Concertées de la Communauté Française de Belgique (ARC grant number 04/09-307). G.L.M. is a fellow from the Fonds Xenophilia, ULB. The laboratory is a member of the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265). The BiGRe laboratory is supported by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet). Funding to pay the Open Access publication charges for this article was provided by the Région Wallonne de Belgique (TransMaze project 415925).

Conflict of interest statement. None declared.

REFERENCES

1. Jeong,H., Mason,S.P., Barabàsi,A.L. and Oltvai,Z.N. (2001) Lethality and centrality in protein networks. *Nature*, **411**, 41–42.
2. Jeong,H., Tombor,B., Albert,R., Oltvai,Z.N. and Barabàsi,A.L. (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.

3. Gagneur,J., Jackson,D.B. and Casari,G. (2003) Hierarchical analysis of dependency in metabolic networks. *Bioinformatics*, **19**, 1027–1034.
4. Croes,D., Couche,F., Wodak,S.J. and van Helden,J. (2006) Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.*, **356**, 222–236.
5. Luscombe,N.M., Babu,M.M., Yu,H., Snyder,M., Teichmann,S.A. and Gerstein,M. (2004) Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, **431**, 308–312.
6. Babu,M.M., Luscombe,N.M., Aravind,L., Gerstein,M. and Teichmann,S.A. (2004) Structure and evolution of transcriptional regulatory networks. *Curr. Opin. Struct. Biol.*, **14**, 283–291.
7. Fukuda,K. and Takagi,T. (2001) Knowledge representation of signal transduction pathways. *Bioinformatics*, **17**, 829–837.
8. Deville,Y., Gilbert,D., van Helden,J. and Wodak,S.J. (2003) An overview of data models for the analysis of biochemical pathways. *Brief Bioinform.*, **4**, 246–259.
9. Huber,W., Carey,V.J., Long,L., Falcon,S. and Gentleman,R. (2007) Graphs in molecular biology. *BMC Bioinform.*, **8 (Suppl 6)**, S8.
10. von Mering,C., Krause,R., Snel,B., Cornell,M., Oliver,S.G., Fields,S. and Bork,P. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403.
11. Sprinzak,E., Sattath,S. and Margalit,H. (2003) How reliable are experimental protein-protein interaction data? *J. Mol. Biol.*, **327**, 919–923.
12. Yook,S.-H., Oltvai,Z.N. and Barabasi,A.-L. (2004) Functional and topological characterization of protein interaction networks. *Proteomics*, **4**, 928–942.
13. Letovsky,S. and Kasif,S. (2003) Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, **19 (Suppl 1)**, i197–i204.
14. Breitkreutz,B.-J., Stark,C., Reguly,T., Boucher,L., Breitkreutz,A., Livstone,M., Oughtred,R., Lackner,D.H., Bähler,J., Wood,V. *et al.* (2008) The biogrid interaction database: 2008 update. *Nucleic Acids Res.*, **36 (Database issue)**, D637–D640.
15. Mewes,H.W., Amid,C., Arnold,R., Frishman,D., Guldener,U., Mannhaupt,G., Munsterkotter,M., Pagel,P., Strack,N., Stumpflen,V. *et al.* (2004) MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.*, **32 (Database issue)**, D41–D44.
16. Janky,R. and van Helden,J. (2008) Evaluation of phylogenetic footprint discovery for predicting bacterial cis-regulatory elements and revealing their evolution. *BMC Bioinform.*, **9**, 37.
17. Scott,J., Ideker,T., Karp,R.M. and Sharan,R. (2006) Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.*, **13**, 133–144.
18. Bebek,G. and Yang,J. (2007) Pathfinder: mining signal transduction pathway segments from protein-protein interaction networks. *BMC Bioinform.*, **8**, 335.
19. Rahman,S.A., Advani,P., Schunk,R., Schrader,R. and Schomburg,D. (2005) Metabolic pathway analysis web service (pathway hunter tool at cubic). *Bioinformatics*, **21**, 1189–1193.
20. van Helden,J., Gilbert,D., Wernisch,L., Schroeder,M. and Wodak,S. (2001) Applications of regulatory sequence analysis and metabolic network analysis to the interpretation of gene expression data. In O.Gascuel and M.-F.Sagot (eds), *Computational Biology : First International Conference on Biology, Informatics, and Mathematics, JOBIM 2000. LNCS Vol. 2066*, Springer, Montpellier, pp. 155–172.
21. van Helden,J., Wernisch,L., Gilbert,D. and Wodak,S.J. (2002) Graph-based analysis of metabolic networks. *Ernst Schering Res. Found.Workshop*, **38**, 245–274.
22. Croes,D., Couche,F., Wodak,S.J. and van Helden,J. (2005) Metabolic pathfinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Res.*, **33 (Web Server issue)**, W326–W330.
23. Jimenez,V. and Marzal,A. (1999) Computing the k shortest paths: a new algorithm and an experimental comparison. *Proc. 3rd Int. Worksh. Algorithm Engineering (WAE 1999)*, **1668**, 15–29.
24. Spirin,V. and Mirny,L.A. (2003) Protein complexes and functional modules in molecular networks. *Proc. Natl Acad. Sci. USA*, **100**, 12123–12128.
25. Han,J.-D.J., Dupuy,D., Bertin,N., Cusick,M.E. and Vidal,M. (2005) Effect of sampling on topology predictions of protein-protein interaction networks. *Nat. Biotechnol.*, **23**, 839–844.
26. Brohée,S. and van Helden,J. (2006) Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinform.*, **7**, 488.
27. Milenkovic,T., Lai,J. and Przulj,N. (2008) Graphcrunch: a tool for large network analyses. *BMC Bioinform.*, **9**, 70.
28. Hu,Z., Ng,D.M., Yamada,T., Chen,C., Kawashima,S., Mellor,J., Linghu,B., Kanehisa,M., Stuart,J.M. and DeLisi,C. (2007) Visant 3.0: new modules for pathway visualization, editing, prediction and construction. *Nucleic Acids Res.*, **35 (Web Server issue)**, W625–W632.
29. Shannon,P., Markiel,A., Ozier,O., Baliga,N.S., Wang,J.T., Ramage,D., Amin,N., Schwikowski,B. and Ideker,T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.
30. Gene Ontology Consortium. (2008) The Gene Ontology project in 2008. *Nucleic Acids Res.*, **36 (Database issue)**, D440–D444.
31. Sharan,R., Ulitsky,I. and Shamir,R. (2007) Network-based prediction of protein function. *Mol. Syst. Biol.*, **3**, 88.
32. Krogan,N.J., Cagney,G., Yu,H., Zhong,G., Guo,X., Ignatchenko,A., Li,J., Pu,S., Datta,N., Tikuisis,A.P. *et al.* (2006) Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature*, **440**, 637–643.
33. Pereira-Leal,J.B., Enright,A.J. and Ouzounis,C.A. (2004) Detection of functional modules from protein interaction networks. *Proteins*, **54**, 49–57.
34. Enright,A.J., Dongen,S.V. and Ouzounis,C.A. (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.*, **30**, 1575–1584.
35. Lattimore,B.S., van Dongen,S. and Crabbe,M.J.C. (2005) Genemcl in microarray analysis. *Comput. Biol. Chem.*, **29**, 354–359.
36. Van Dongen,S. (2000) Graph clustering by flow simulation. *Ph.D. Thesis.* Centers for Mathematics and Computer science (CWI), University of Utrecht.
37. King,A.D., Przulj,N. and Jurisica,I. (2004) Protein complex prediction via cost-based clustering. *Bioinformatics*, **20**, 3013–3020.
38. Lima-Mendez,G., van Helden,J., Toussaint,A. and Leplae,R. (2008) Reticulate representation of evolutionary and functional relationships between phage genomes. *Mol. Biol. Evol.*, **25**, 762–777.
39. Gavin,A.-C., Aloy,P., Grandi,P., Krause,R., Boesche,M., Marzioch,M., Rau,C., Jensen,L.J., Bastuck,S., Dümpelfeld,B. *et al.* (2006) Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**, 631–636.
40. von Mering,C., Jensen,L.J., Kuhn,M., Chaffron,S., Doerks,T., Krüger,B., Snel,B. and Bork,P. (2007) String 7-recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res.*, **35 (Database issue)**, D358–D362.
41. Uetz,P., Giot,L., Cagney,G., Mansfield,T.A., Judson,R.S., Knight,J.R., Lockshon,D., Narayan,V., Srinivasan,M., Pochart,P. *et al.* (2000) A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, **403**, 623–627.
42. Ito,T., Chiba,T., Ozawa,R., Yoshida,M., Hattori,M. and Sakaki,Y. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl Acad. Sci. USA*, **98**, 4569–4574.

Network Analysis Tools: from biological networks to clusters and pathways

Sylvain Brohée, Karoline Faust, Gipsi Lima-Mendez, Gilles Vanderstocken & Jacques van Helden

Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe), Université Libre de Bruxelles, Campus Plaine, CP 263, Boulevard du Triomphe, B-1050 Bruxelles, Belgium. Correspondence should be addressed to J.v.H. (Jacques.van.Helden@ulb.ac.be).

Published online 18 September 2008; doi:10.1038/nprot.2008.100

Network Analysis Tools (NeAT) is a suite of computer tools that integrate various algorithms for the analysis of biological networks: comparison between graphs, between clusters, or between graphs and clusters; network randomization; analysis of degree distribution; network-based clustering and path finding. The tools are interconnected to enable a stepwise analysis of the network through a complete analytical workflow. In this protocol, we present a typical case of utilization, where the tasks above are combined to decipher a protein–protein interaction network retrieved from the STRING database. The results returned by NeAT are typically subnetworks, networks enriched with additional information (i.e., clusters or paths) or tables displaying statistics. Typical networks comprising several thousands of nodes and arcs can be analyzed within a few minutes. The complete protocol can be read and executed in ~1 h.

INTRODUCTION

This is the last article in a series of four protocols for the analysis of regulatory sequences with the Regulatory Sequence Analysis Tools[1] (http://rsat.ulb.ac.be/rsat/) and biological networks with the Network Analysis Tools (NeAT)[2] (http://rsat.ulb.ac.be/neat/). The first article[3] presents a protocol to predict the location of binding sites for transcription factors whose specificity is already known (pattern matching). In the second article[4], we describe a protocol for the *ab initio* discovery of biological signals in biological sequences (pattern discovery). The third article[5] shows how to write scripts to automate the analysis on multiple clusters of genes using Web services. In this article, we describe a workflow for deciphering biological networks by combining network comparison, module identification and path finding. This protocol can be executed independently of the three other ones.

Network biology is emerging as a new field in biology, due to the increasing availability of genome-scale data sets of molecular interactions, such as those resulting from high-throughput technologies (e.g., protein interactions, regulatory networks and metabolome). Extracting relevant information from this huge amount of data requires dedicated tools. These data sets are commonly represented as graphs (or networks), where nodes represent molecules, and arcs their interactions. This representation eases data integration and makes it possible to apply well-known network algorithms to analyze the data.

In this protocol, we show how large biological networks can be explored by combining a set of modular tools accessible via a Web interface named NeAT. We describe hereafter the typical questions that can be addressed.

Network topology. It has been observed that some topological properties distinguish biological networks from random networks[6,7]. Noticeably, the distribution of degree (the number of arcs connected per nodes) is often claimed to follow a power-law distribution[6,8]. The tool ***graph-topology*** can be used to analyze the degree distribution of any kind of network.

Network comparison. Given two networks (i.e., protein–protein interaction networks from two different experiments), one would like to analyze their degree of overlap. This question is answered by the tool ***compare-graphs***, which computes the intersection, union and difference between two networks and estimates the statistical significance of the overlap.

Node neighborhood. Given a protein, gene or another node in a biological network, it is of interest to identify its direct or indirect neighbors in this network. This is the task of the tool ***graph-neighbours***, which returns the neighbors of a query node in a network up to a certain distance. This tool can be applied for instance on protein–protein interaction networks to retrieve the interaction partners of a given protein.

Cluster analysis. Various algorithms have been implemented to extract clusters (i.e., groups of densely connected nodes) from biological networks[9–13]. Among those, ***MCL***[14] algorithm has been shown to obtain good performances for extracting protein complexes from protein interaction networks[15–17]. In addition, this algorithm can deal with large graphs and is very efficient in time. For these reasons, we included MCL in the NeAT suite. The clusters resulting from MCL or other methods can be compared to some reference groups (e.g., functional classes) with the program ***compare-classes*** and mapped onto networks with ***graph-get-clusters***. Upon partitioning with MCL, each node belongs to only one cluster. However, sometimes the assignment of a node to a single cluster is an over-simplification of the biological data, for example, a protein may be part of different protein complexes. In those cases, it would be better to describe how much each node is related to the different clusters. The program ***graph-cluster-membership*** postprocesses a clustering result and calculates the membership as the proportion of edges (or weight) linking each node to each cluster. The node–cluster relationships are described as a membership matrix, where each row represents a node and each column a cluster.

Path finding. Given a biological network and two nodes of interest, a common task is to find a biological meaningful path connecting those nodes in the network. For instance, path-finding algorithms are applied to uncover signal transduction or metabolic pathways in protein–protein interaction or metabolic networks, respectively[18–21]. Recently, we evaluated the performance of a *k*-shortest path-finding algorithm for metabolic pathway inference and found high accuracies if appropriate weights are set on the network[22].

Network randomization. Negative controls are essential to estimate the relevance of the results. The tool ***random-graph*** proposes different procedures to randomize a network, which can then be submitted to the same workflows as the original network.

Network alteration. To test the robustness of analytic methods to the presence of noise, or to the incompleteness of information, the tool ***alter-graph*** allows to modify an existing network by random addition or deletion of nodes and/or edges.

Network display. NeAT includes a tool called ***display-graph***, which generates static images of the input networks. Such drawings are convenient for a quick inspection of the results from the Web browser. For more sophisticated layouts and for a dynamical manipulation of the drawing, we recommend graph editors such as yEd (http://www.yworks.com/en/products_yed_about.html) or Cytoscape[23]. The tool ***convert-graph*** permits to export any network analyzed with NeAT into Graph Modeling Language (GML) (http://www.infosun.fim.uni-passau.de/Graphlet/GML/gml-tr.html), a file format supported by both editors.

Figure 1 depicts the way in which the NeAT can be connected. We suggest the reader to follow this flow chart progressively during the execution of the protocol.

Comparison to other graph analysis tool suites

A large variety of graph analysis tools exist. We may classify them in three categories: (i) libraries that can only be used programmatically, for example, Boost (http://www.boost.org/), igraph (http://cneurocvs.rmki.kfki.hu/igraph/index.html) or JUNG (http://jung.sourceforge.net/); (ii) stand-alone tools with graphical user interface (GUI) (Pajek[24], Network Workbench (http://nwb.slis.indiana.edu), BiologicalNetworks[25], VisANT[26], yEd, Cytoscape, etc.) and (iii) tools with GUI available via the Web such as tYNA (http://tyna.gersteinlab.org/tyna/) or CABiNeT (http://mips.gsf.de/genre/proj/CABiNeT/).

Usually, the libraries offer generic graph algorithms, whereas stand-alone or Web-based tool suites are often specialized. For instance, VisANT, Cytoscape (with its plugins) and BiologicalNetworks focus on the analysis and display of biological networks, whereas yEd offers a flexible interface for the display, layout and edition of general-purpose graphs, but is equipped with limited analysis functions. Pajek and Network Workbench are stand-alone tools for generic graph analysis. Cytoscape (with plugins) and BiologicalNetworks allow in addition retrieval and integration of biological networks.

We describe hereafter some of the advantages and current limitations of NeAT.

Main advantages. The main advantages of NeAT are

(1) NeAT supports a variety of modular tools, which can either be used separately, or combined in a workflow. These tools include a number of unique features (fuzzy clustering, Web access to MCL and RNSC, *k*-shortest paths with multiple start and end nodes, statistical comparison of classes and clusters, etc.) that are currently not available in other packages.

(2) The programs are designed to enable treating very large graphs (several thousands of nodes) without excessive cost in memory or time.

(3) Although most of the analyses can also be performed in specialized software packages such as R, the NeAT Web site offers a user-friendly access for biologists who are not familiar with programming languages.

(4) NeAT can be run on command line, either by installing it locally or by calling Web services. This is not the case of the other stand-alone and Web-based tools (a notable exception is Pajek). The programmatic access (either as stand-alone application or as Web services) allows one to automate the executions of workflows for multiple data sets, which would require hundreds or thousands of manual operations with conventional GUIs or on a Web site. To our knowledge, there is only one other network tools suite enabling workflows, namely tYNA. NeAT and tYNA are complementary: NeAT supports path-finding, graph-based clustering (MCL, RNSC and fuzzy clustering), network randomization and cluster comparisons, whereas tYNA includes tools to find motifs in networks. Because both tools support Web services, they can be easily combined in workflows, either by programming client scripts or using GUIs such as Taverna[27].

(5) NeAT may be used for any kind of network, but it was developed with biological networks in mind. The tools have been extensively tested on a variety of biological networks (protein–protein interaction networks[17], evolutionary networks[28] and metabolic networks[21,22]). Extensive evaluation is rarely reported for other biological network tools suites.

Main limitations. NeAT essentially provides facilities for the analysis of networks, clusters and pathways, but is not focused on

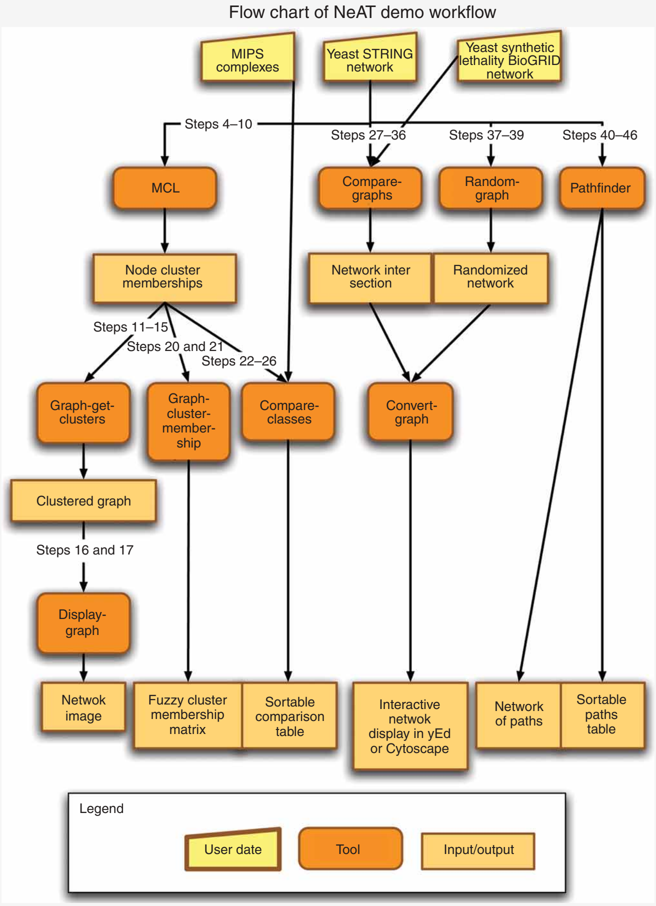


Figure 1 | Flow chart of the data, tools and results described in this protocol. Yellow represents the data set, orange the tool and light brown the results.

the problem of network visualization. This limitation, however, is easily circumvented by installing some specialized visualization software: all graphs generated by NeAT can be exported to several formats, including GML, which can be loaded with Cytoscape, yEd and VisANT, and DOT, which can be loaded with Graphviz.

To summarize, NeAT addresses the needs of researchers interested in the analysis of biological networks. Some tools may require background knowledge (e.g., MCL, fuzzy clustering), whereas others are intuitive and easy to use (e.g., graph conversion and comparison).

For the user with experience in programming, NeAT can be run on command line or within workflow management environments such as Taverna. Otherwise, the user may access NeAT via its Web interface, guided by tutorials and demos. To our knowledge, no other biological network tools suite exists that combines all the features of NeAT.

Other applications of this protocol

For the sake of consistency, the cases treated in this protocol are restricted to protein interaction networks. The tools available in NeAT can also be used to analyze other types of biological networks representing other types of interactions, for example, regulation, signal transduction, metabolic reactions, and ecology. The fuzzy clustering approach was initially conceived to address the problem of classifying phage genomes while taking into account the frequent exchanges of genetic material between them[28]. The *k*-shortest path-finding algorithm has previously been applied to infer relevant pathways in metabolic networks[29,30].

MATERIALS

EQUIPMENT

- This protocol describes an online tool. The only requirement is a computer with Internet connection. Optionally, you can install yEd (http://www.yworks.com/) or Cytoscape[23] for visualization
- Sample interaction networks, which can be obtained from various biological databases. As examples, we cite the following:
  - STRING[31] (http://string.embl.de/), a database integrating seven different types of evidences for physical and/or functional interactions between proteins: experimental evidences, phylogenetic profiles ('co-occurrence'), gene fusion/fission, synteny ('neighborhood'), coexpression, text mining and a data set called 'database', regrouping several criterion selected by the STRING annotation team
  - BioGRID[32] (http://www.thebiogrid.org/), a database of protein and genetic interactions including >116,000 curated interactions from yeast, *Caenorhabditis elegans*, drosophila and human
  - BioCyc[33] (http://www.biocyc.org/) or KEGG[34] (www.genome.jp/kegg/), the two main metabolic pathway databases
- The data required for the study cases treated in this protocol is available in the data repository site: http://rsat.ulb.ac.be/nedt/. All the networks used to illustrate this protocol were taken from the yeast *Saccharomyces cerevisiae*. We selected various networks representing diverse types of interactions between biological molecules (protein interactions, metabolism, protein complexes, genetic interactions, etc.)
  - Protein–protein interactions (physical and functional). From the STRING database[31], we extracted a subset labeled as 'database' by the STRING team. Under this label, they regrouped different types of protein–protein interactions and metabolic relationships (Jensen L., personal communication). This network contains 1,237 nodes representing proteins, and 11,027 edges representing a mixture of physical and functional protein–protein interactions. The interactions between two proteins are considered symmetrical; it is an undirected graph. The network is stored in data repository, a tab-delimited text file named *yeast_string_database_graph_names_undirected.tab*
  - The ***synthetic lethality network*** was extracted from the BioGRID database[32]. It represents genes (2,353 nodes) whose individual deletion is viable, but whose paired deletions (12,419 edges) are lethal
  - Protein complexes. The file *mips_complexes_names.tab* describes the collection of protein complexes annotated in the MIPS database[35]. Complexes detected only by high-throughput experiments were discarded from the data set. The first column of the file gives the gene name, the second column the complex name and the third column the gene identifier. In total, the file contains 1,121 distinct proteins forming 243 distinct complexes. Note that a protein can belong to several complexes
  - Signal transduction pathway. As study case for the path finding, we take a yeast signal transduction pathway mentioned by Scott and colleagues[18]. This pathway, known to regulate filamentous growth in yeast, starts with RAS2 and ends with TEC1. The authors attempt to recover this pathway with a path-finding algorithm based on color coding[18]. We will try to recover it using *Pathfinder*
- Incompatibility between file formats is a constant problem in bioinformatics. To facilitate the use of the Web site, most tools support several among the most popular formats used to describe networks. A description of the supported format is given in **Box 1**

BOX 1 | MAIN GRAPH FORMATS

The ***tab-delimited format*** is a convenient and intuitive way to encode a graph. Each row represents an edge and each column an attribute of this edge. The two column fields are the source and target nodes. If the graph is directed, the source node is the node from which the arc leaves and the target node is the node to which the arc arrives. Logically, in undirected graph, the columns containing the source and the target nodes may be swapped. Orphan nodes can be included by specifying a source node without target. Some additional edge attributes (weight, label, color) can be placed in additional columns. The tool *Pathfinder* extends this format by supporting any number of attributes on nodes or edges. Check the *Pathfinder* help page for more details.

The ***GML format*** allows for the specification of additional layout and display attribute, such as node position, as well as the color, the label and the width of nodes and edges. A Graph Modeling Language (GML) file is made up of nested key-value pairs. The most popular graph editors (such as Cytoscape and yEd) support GML as input format.

The ***DOT format*** is a plain text graph description language. DOT files can be loaded in the programs of the suite Graphviz (http://www.graphviz.org/). It is a simple way of describing graphs in a human- and computer-readable format. Similar to GML, DOT supports various attributes on nodes (i.e., color, width, label).

Several tools also accept adjacency matrices as input. An ***adjacency matrix*** is an $N \times N$ table (with N the number of nodes), where a cell $A[i,j]$ indicates the weight of the edge between nodes i and j (or 1 if the graph is unweighted).

The Network Analysis Tools program ***convert-graph*** facilitates the handling of these formats by supporting interconversions between various input (tab, gml, adjacency) and output formats (tab, dot, gml, adjacency).

PROCEDURE

Downloading a sample network

1| We will show on an example workflow how the different tools of NeAT can be combined to analyze a network taken from the STRING database. Open a connection to the data repository for this protocol (see EQUIPMENT).

2| Download the network file *yeast_string_database_graph_names_undirected.tab* on your computer. It is described in a tab-delimited file that contains five columns. Each row represents one interaction between two genes or between their products. As described in **Box 1**, the two first columns indicate the name of the ***Source*** and ***Target*** genes/proteins of the source and target nodes. The third column contains a score ranging from 0 to 900, which reflects the reliability of the indications available for this interaction. Higher scores represent more reliable interactions. In this case, the score is higher if an interaction is found several times in different data set. The columns 4 and 5 contain the gene identifiers corresponding to the gene names in columns 1 and 2.

3| Open a connection to the NeAT Web server: http://rsat.ulb.ac.be/neat/.

Cluster analysis

4| *Extracting clusters from the network with MCL (Steps 4–10)*. We will first apply graph-based clustering to detect groups of highly interconnected nodes in the sample network. For this, we will use the MCL algorithm, a fast unsupervised clustering algorithm based on simulation of flows in graphs[14]. In the menu from the left panel, click on the link ***MCL clustering*** to open the MCL query form.

5| Click on the ***Browse...*** button, and choose the file containing the network (e.g., *yeast_string_database_graph_names_undirected.tab* for the study case discussed here).

6| Specify the columns containing the source, target and (optionally) weight attributes of the tab-delimited file. In our example file, the source and target columns are by default 1 and 2 so we only have to add the weight column: ***Weight column*** = 3. Note that if you want to work with the gene identifiers instead of the gene names, you could have used value 4 and 5 in the source and target column fields. However, this is not recommended in this protocol as in the following we will only work with gene names.
▲ CRITICAL STEP The weighting of edges strongly affects the MCL result, because the principle of the algorithm is to iteratively enforce the weight of the most 'important' edges in the network. The 'importance' of an edge is determined by both its initial weight and its place in the network.

7| Choose an inflation value (between 1.2 and 5). For the study case, select 1.8.
▲ CRITICAL STEP This parameter acts on the granularity of the clustering procedure, that is, the number of clusters (and consequently the number of elements per cluster). The number of clusters increases with the inflation value. This parameter must thus be fine-tuned according to the structure of the network. In a recent evaluation, we found that an inflation value of 1.8 was optimal for protein–protein interaction networks[17].

8| Click on the ***GO*** button. The processing should take a bit less than one 1 min.
? TROUBLESHOOTING

9| The result page displays a figure showing the cluster size distribution, that is, the number of clusters (ordinate) of each size (abscissa). The page also contains a link to the clustering result. This file can be saved by right-clicking on the URL link and selecting ***Save link as...***, and save it on your computer under the name *yeast_string_MCL_clusters.tab*.

10| To inspect the result file, you can either use a text editor to open the file *yeast_string_MCL_clusters.tab* stored on your computer, or click on the URL on the result page. The MCL result is a simple two-column table, where the first column indicates the node names (gene names in our case), and the second column the cluster names. A quick inspection of this table from top to bottom shows that the first clusters contain more nodes than the last ones. MCL sorts the clusters by decreasing order of size.

11| *Extracting the subnetwork defined by the clusters (Steps 11–17)*. We will now map the clusters resulting from MCL onto their original network. For this, there are two alternative ways to proceed: directly load the files stored on the server (option A) or transfer the network and cluster files from your computer (option B).

(A) Directly load the files stored on the server

(i) The MCL result page displays a '***Next step***' box, allowing you to send the MCL output to several alternative tools. Click the button ***Map those cluster on the network***. This will call a form *graph-get-clusters*, with prefilled values for the parameters ***Graph*** and ***Clusters***.

(B) Transfer the network and cluster files from your computer

(i) An alternative way to enter data in the tool *graph-get-clusters* is to click on the link '***Map clusters onto network***' in the left menu. This will open an empty form, in which you will have to enter the data (for our study case, the graph is in the file *yeast_string_database_graph_names_undirected.tab* and the clusters in the file *yeast_string_MCL_clusters.tab*). However, this would require to transfer those two files to the server, albeit it already contains a copy of both in the temporary directory. Whenever possible, you should thus use the '***Next step***' buttons rather than transferring the files back and forth between your computer and the server.

12| The main choice for the tool *graph-get-clusters* is the ***output type***. The program supports two types of operations between the network and the clusters. (i) The option ***annotated graph*** labels each edge according to its intracluster or intercluster nature. (ii) ***intracluster edges*** selects a subnetwork restricted to the intracluster edges (intercluster edges are simply deleted from the network). You can experiment the three options. In this section of the protocol, we will extract the subnetwork defined by the MCL clusters. For this, select the option ***intracluster edges***.

13| Several output formats are proposed, but for the visualization purpose, select the intracluster edges output in the ***GML format***.

14| Click on the button ***GO***.

15| The result page should appear after <1 min, displaying a set of buttons for postprocessing the graph-get-cluster result, and a link toward the result file. You can store the resulting GML graph on your computer for later use by right-clicking on the URL in the *graph-get-clusters* result page. Save the result in a file named *yeast_string_MCL_intra_cluster.gml*.

16| A quick way to visualize the result is to fetch it to the NeAT visualization tool. However, beware that this tool offers limited functionalities: it returns a static image, with a simplistic layout. The main function of this tool is to provide a quick view of the result, before visualizing it with specialized tools. To visualize the result network with NeAT, click on the button ***Display the graph***. A new form will then be displayed. Select the desired output format. If the network is weighted (e.g., our study case), you can activate the option ***Edge width proportional to the weight***. To obtain the figure, click on the button ***GO***. This process may be slow (>1 min) depending on the size of the graph.

17| For a better visualization of the network, open the GML formatted file obtained in Step 15 with Cytoscape, yEd or any other visualization program of your choice. After having opened the GML file, you need to apply some layout to display nodes and edges harmoniously. For yEd and Cytoscape we recommend the option ***Organic layout***. After this, you should see a set of well-separated components, each corresponding to a MCL cluster. Each cluster is displayed with a specific color for the edges (**Fig. 2a**).

? TROUBLESHOOTING

18| *Mapping the clusters onto the network (Steps 18 and 19)*. In the previous section, we used ***graph-get-clusters*** to separate the MCL clusters by deleting intercluster edges from the original network. Alternatively, the same tool can be used to label all the edges according to the cluster composition. Come back to Step 12, but this time, select '***annotated graph (all edges)***' as ***output type***.

19| Repeat Steps 13–17, and compare visually the result with that obtained in the previous section (**Fig. 2b**).

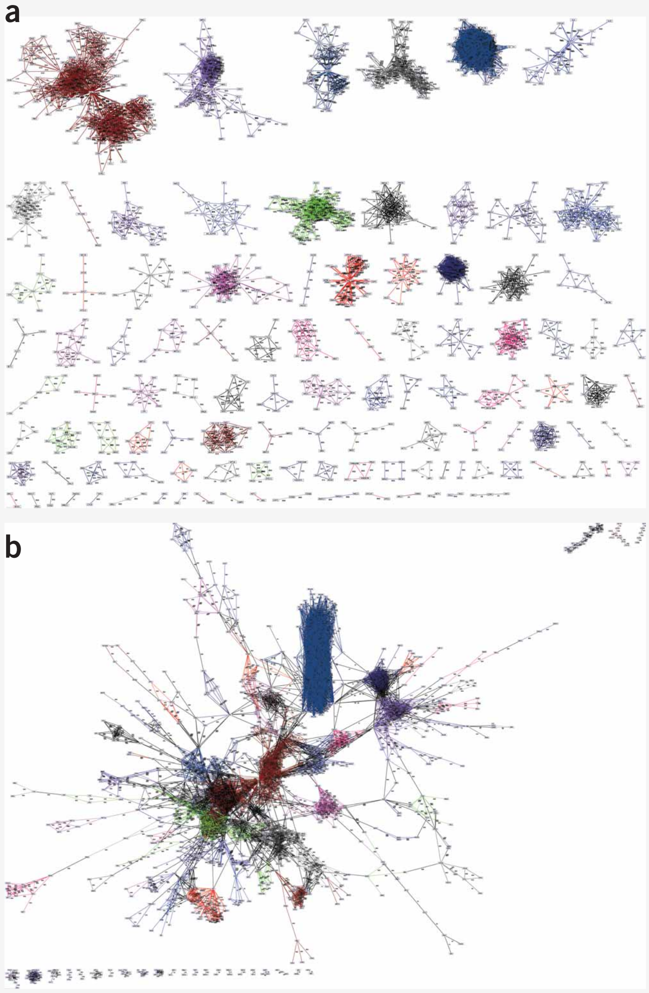


Figure 2 | Mapping of the clusters obtained with the MCL algorithm on the STRING database data set. (**a**) Only the intracluster edges were conserved and each cluster is highlighted with a different color. (**b**) Intercluster (black) and intracluster (colored) edges are both displayed. The layout and display were obtained with yEd.

Fuzzy clustering

20| We will now use the tool ***graph-cluster-membership*** to compute the degree of membership of each node to each of the cluster obtained from MCL. This can be done in either of two ways. (i) Come back to the page with the MCL output and click on '***Cluster membership***' to open ***graph-cluster-membership***. (ii) Alternatively, you can click on the link ***Cluster membership*** in the left panel, and specify the graph parameters as in Steps 5 and 6. To upload the MCL output, click on the button besides '***Upload clusters from file***': and specify the location of the file *yeast_string_MCL_clusters.tab* on your disk.

21| Search for the membership matrix and select weight as stat.
▲ CRITICAL STEP For weighted graphs, weight or relative weight may be chosen. Otherwise, the strength of the links is not considered for calculating the membership of a node to a cluster. When relative weight or relative edge is selected, the weights or number of edges of a node to a cluster are divided by the number of nodes of that cluster.

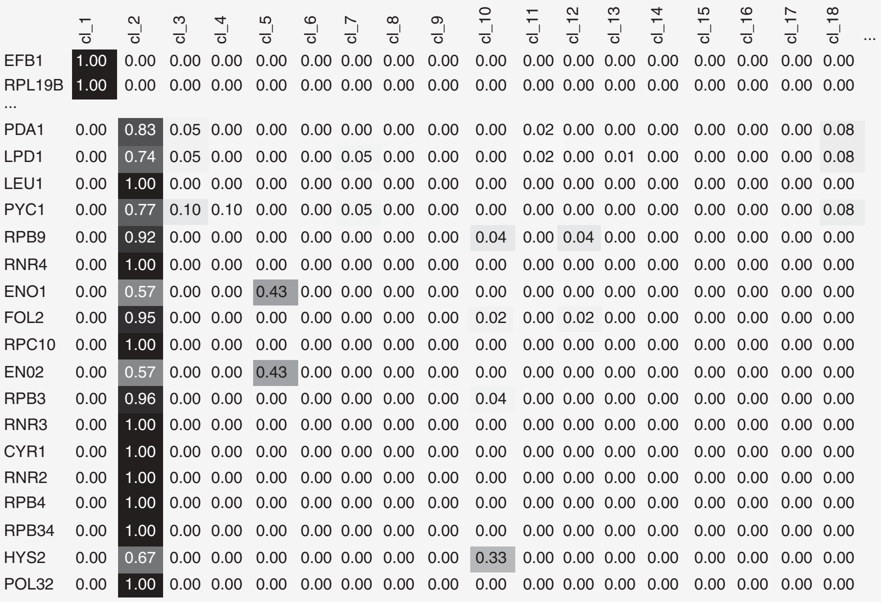
22| Click on the ***GO*** button. After a minute, a page appears with links to three files: a tab-delimited text file, and two image files providing, respectively, low- and high-resolution heatmaps. In all cases, the output displays the membership matrix, where entries correspond to the membership degree of the node given by the row to the cluster given by the column. The text-formatted table contains the numeric values of the memberships coefficient associating each node (row) to each cluster (column). This is a tab-delimited file that can be loaded in various programs (e.g., Excel, R) for further processing. The heatmap is a graphical representation of the same data, where the gray level is proportional to the degree of membership (**Fig. 3**).
? TROUBLESHOOTING

23| *Comparing the clusters with reference classes (Steps 23–27)*. To evaluate the biological relevance of the clusters discovered with MCL, we can compare them with some reference classification, for example the Gene Ontology[36] or the collection of protein complexes from the MIPS database. To illustrate this, we will compare the MCL clusters obtained above with the complexes stored in the MIPS database. Each MCL cluster will be compared to each complex of the database. Cluster/complex comparisons will then be scored with different statistics described in the manual page of the tool and in **Box 2**. Come back to the page with the MCL result (Step 9). On in the ***Next step*** box, click on the button '***Compare these clusters with other clusters***'. Alternatively, in case you saved the MCL result in a file, you can directly click on the link ***Compare clusters/classes*** in the left panel and upload the MCL result file with the ***Browse*** button under ***query class***.

24| As reference classes, we will use the collection of MIPS complexes. For this, first download the file *mips_complexes_names.tab* from the data repository (see EQUIPMENT).

25| We will now specify the ***reference classes*** in the *compare-classes* form. To indicate that MIPS complexes will serve as reference classes, click on the ***Browse...*** button below ***Reference classes***, and select the file ***mips_complexes_names.tab*** that you downloaded on your computer at Step 24. In NeAT, classes are described with the same tab-delimited format as clusters: each row describes the membership of one element (first column) to a class (second column). Optionally, an additional column can be specified with the option '***Score column***', to indicate a score that will be used to compute some similarity metrics (e.g., dot product). For this study case, the MIPS complexes are described in a three-column file indicating the protein name (first column), the complex (second column) and the gene ID (third column). There is no score associated to the protein-complex membership, and we will thus leave empty the option ***score column***.

26| The options ***Thresholds on return fields*** (see **Box 2**) allow one to combine various constraints to select the most significant intersections between query and reference clusters/classes. The default threshold on significance (sig ≥ 0) usually gives satisfactory results. For our study case, the other thresholds do not need to be changed.



	cl_1	cl_2	cl_3	cl_4	cl_5	cl_6	cl_7	cl_8	cl_9	cl_10	cl_11	cl_12	cl_13	cl_14	cl_15	cl_16	cl_17	cl_18	...
EFB1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
RPL19B	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
...																			
PDA1	0.00	0.83	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.08	
LPD1	0.00	0.74	0.05	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.08	
LEU1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
PYC1	0.00	0.77	0.10	0.10	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	
RPB9	0.00	0.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	
RNR4	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
ENO1	0.00	0.57	0.00	0.00	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
FOL2	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	
RPC10	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
EN02	0.00	0.57	0.00	0.00	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
RPB3	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
RNR3	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
CYR1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
RNR2	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
RPB4	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
RPB34	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
HYS2	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
POL32	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

Figure 3 | Fuzzy-clusters obtained by combining MCL and the graph-cluster-membership tools. The section of the heatmap shows that several proteins have cluster membership percentages larger than zero for several clusters; for example, ENO1 belongs to both clusters cl_2 (57%) and cl_5 (43%).

BOX 2 | METRICS FOR COMPARISONS BETWEEN CLASSES OR BETWEEN GRAPHS

In several sections of this protocol, we try to detect significant intersections between two classifications (e.g., MCL clusters, MIPS complexes, etc.) or between two graphs (e.g., interactome). The Network Analysis Tools suite includes specialized programs to compare classes/clusters (*compare-classes*), or graphs (*compare-graphs*), using various comparison statistics.

In both cases (classes or graphs), we can consider that we have a finite set of N elements. For *compare-classes*, N is the total number of elements that can be a member of any reference or query class (e.g., all the yeast genes). For *compare-graphs*, the N elements are all the edges that could possibly be traced between any pair of nodes of the input graph (e.g., all possible intersections between any pair of proteins).

Let us then define

N the total number of elements in the universe (cluster/class members for *compare-classes*, graph edges for *compare-graphs*);
R a reference set (one class/cluster, or one graph), containing Nr elements;
Q a query set (one class/cluster, or one graph), containing Nq elements;
C the intersection between a query and the reference set;
Nc the number of elements in this intersection.

Maximal number of edges in a graph

The maximal number of arcs between a set of X nodes depends on whether this graph is directed or not and on whether it does or does not admit self-loops. We can easily compute the value in the four possible cases.

Directed	Self-loops	Number of edges
Yes	Yes	$N = X^2$
Yes	No	$N = X^2 - X = X(X-1)$
No	No	$N = \frac{X(X-1)}{2}$
No	Yes	$N = \frac{X + X(X-1)}{2} = \frac{X(X+1)}{2}$

The column "Number of edges" corresponds to the N used for the statistics on graph comparisons.

Jaccard coefficient

The Jaccard coefficient is defined as the ratio between the intersection and the union between two sets.

$$J = \frac{R \cap Q}{R \cup Q} = \frac{Nc}{Nr + Nq - Nc}.$$

The advantage of the Jaccard coefficient is that it gives us an intuitive perception about the mutual coverage of the query and the reference. However, it presents the weakness to be independent of the absolute sizes of the union and intersection. For example, an intersection of 1 element between a set of 3 and a set of 2 elements will give the same Jaccard coefficient as an intersection of 100 between a set of 300 and a set of 100 elements, whereas the random expectation for these two events is very different. A more reliable statistics is the hypergeometric coefficient, as discussed below.

The hypergeometric probability

The hypergeometric distribution is often used to estimate the significance of the intersection between two random selections in a set. The classical example of application of the hypergeometric distribution is the random selection without replacement in an urn containing a set of white and black balls.

The reference set (classes or graph) can be assimilated to the black balls of the urn example. The query set corresponds to the selection without replacement (indeed, a member cannot appear several times in the same class and an edge cannot appear several times in the same graph). The hypergeometric P value indicates the probability to observe by chance at least x elements at the intersection between the query set and the reference set.

$$P_{\mathrm{val}} = P(X \geq Nc) = \sum_{i=Nc}^{Nq} \frac{C_{Nr}^{i} C_{N-Nr}^{Nq-i}}{C_{N}^{Nq}}.$$

The P value can be interpreted as the probability for one comparison to return a false positive.

In the case of *compare-classes*, an important correction has to be done for multitesting. Indeed, each class of the query set (e.g., MCL clusters) will be compared to each class of the reference set (e.g., MIPS complexes). The number of comparisons is thus the product between the number of classes in the query set, and in the reference set, respectively. Thus, the nominal P value can be misleading because even a low P value is expected to emerge by chance when the number of comparisons is very high. A classical correction for this multitesting is to compute the E value.

$$\mathrm{E}_{\mathrm{val}} = T \cdot P_{\mathrm{val}},$$

where T is the number of tests. The E value represents the number of false positives to be expected in a battery of t-tests.

To give a realistic order of magnitude, in our study case, we compared 243 clusters obtained from MCL with 107 complexes annotated in the MIPS. The number of comparisons is thus $T = 243 \times 107 = 26{,}001$. Consequently, with an upper threshold of 1% on the P value, we would expect at least 260 false positives!

It is also usual to perform a minus log conversion of the E value, which gives the 'significance score'.

$$\mathrm{sig} = -\log_{10}(\mathrm{E}_{\mathrm{val}}).$$

The *sig* score gives an intuitive perception of the exceptionality of the intersection between sets: a negative significance indicates that an intersection of at least that size is likely to occur by chance, a positive value means that it is significant.

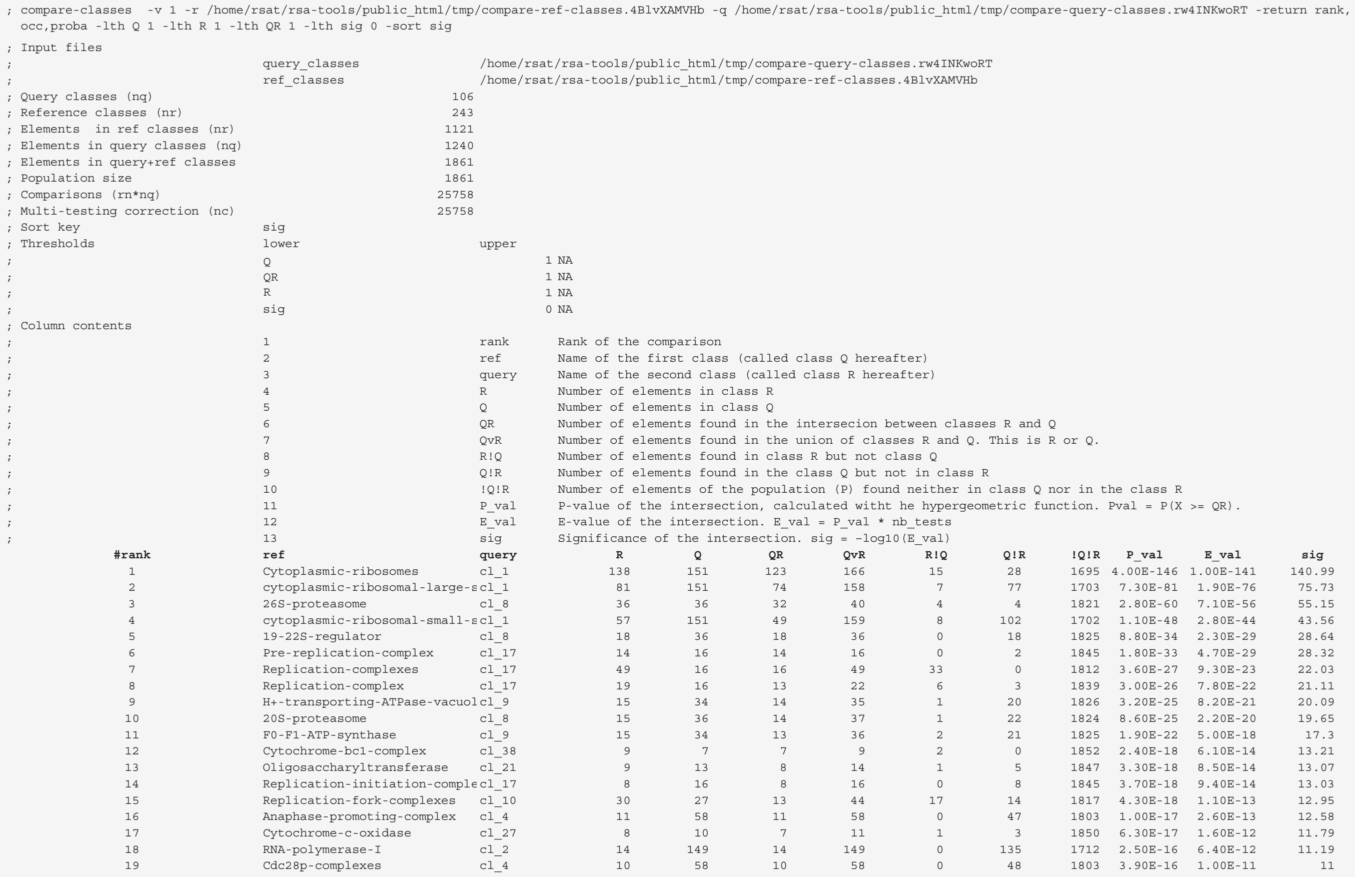
27| Click on the ***GO*** button. After a few seconds, a result page appears with links pointing toward two alternative output formats: tab-delimited text file or hypertext markup language (HTML) page (**Fig. 4**). The tab-delimited text file can be downloaded to your computer and then imported to various applications for further analysis. The HTML format is useful for inspecting and handling the result on the Web browser. The NeAT HTML tables support dynamic reordering of the rows according to the values of any column, by clicking on the header of this column. The first line of the result file indicates the parameters used for the analysis and documents the content of the columns. This is followed by a result table, where each line reports the comparison between one MCL cluster and one complex.

Network comparison

28| In this section, we will use the tool ***compare-graphs*** to compare the interactions annotated for the yeast *Saccharomyces cerevisiae* in the STRING database with the synthetic lethality relationships obtained from the BioGRID database. Download on your computer the file *Saccharomyces_cerevisiae_biogrid_synthetic_lethality_names.tab* from the data repository (see EQUIPMENT).

29| In the NeAT menu of the left panel, click on the link ***Network comparison***. For our study case, select as ***Query graph*** the previously downloaded file *yeast_string_database_graph_names_undirected.tab*, and as ***Reference graph*** the file *Saccharomyces_cerevisiae_biogrid_synthetic_lethality_names.tab*, by clicking on the corresponding ***Browse...*** buttons. For each file, you need to specify the columns containing source and target nodes, respectively. In both files, the first column contains the source and second column the target. We thus just have to fill the weight column for the query graph (weight = 3) as done previously. The reference network (synthetic lethality) does not contain edge weights.

30| Since in our example, only the edges of the STRING network are weighted, select ***weight/label of the query*** for the option ***Weight/label on the edges of the output graph***.



```
; compare-classes  -v 1 -r /home/rsat/rsa-tools/public_html/tmp/compare-ref-classes.4BlvXAMVHb -q /home/rsat/rsa-tools/public_html/tmp/compare-query-classes.rw4INKwoRT -return rank,
  occ,proba -lth Q 1 -lth R 1 -lth QR 1 -lth sig 0 -sort sig
; Input files
;                                query_classes                /home/rsat/rsa-tools/public_html/tmp/compare-query-classes.rw4INKwoRT
;                                ref_classes                  /home/rsat/rsa-tools/public_html/tmp/compare-ref-classes.4BlvXAMVHb
; Query classes (nq)                                     106
; Reference classes (nr)                                 243
; Elements  in ref classes (nr)                         1121
; Elements in query classes (nq)                        1240
; Elements in query+ref classes                         1861
; Population size                                       1861
; Comparisons (rn*nq)                                  25758
; Multi-testing correction (nc)                        25758
; Sort key                       sig
; Thresholds                     lower                        upper
;                                Q                                    1 NA
;                                QR                                   1 NA
;                                R                                    1 NA
;                                sig                                  0 NA
; Column contents
;                                1                            rank      Rank of the comparison
;                                2                            ref       Name of the first class (called class Q hereafter)
;                                3                            query     Name of the second class (called class R hereafter)
;                                4                            R         Number of elements in class R
;                                5                            Q         Number of elements in class Q
;                                6                            QR        Number of elements found in the intersecion between classes R and Q
;                                7                            QvR       Number of elements found in the union of classes R and Q. This is R or Q.
;                                8                            R!Q       Number of elements found in class R but not class Q
;                                9                            Q!R       Number of elements found in the class Q but not in class R
;                                10                           !Q!R      Number of elements of the population (P) found neither in class Q nor in the class R
;                                11                           P_val     P-value of the intersection, calculated witht he hypergeometric function. Pval = P(X >= QR).
;                                12                           E_val     E-value of the intersection. E_val = P_val * nb_tests
;                                13                           sig       Significance of the intersection. sig = -log10(E_val)
              #rank              ref                          query              R          Q         QR        QvR        R!Q        Q!R       !Q!R    P_val      E_val        sig
                1                Cytoplasmic-ribosomes        cl_1             138        151        123        166         15         28       1695 4.00E-146 1.00E-141     140.99
                2                cytoplasmic-ribosomal-large-scl_1              81        151         74        158          7         77       1703  7.30E-81  1.90E-76      75.73
                3                26S-proteasome               cl_8              36         36         32         40          4          4       1821  2.80E-60  7.10E-56      55.15
                4                cytoplasmic-ribosomal-small-scl_1              57        151         49        159          8        102       1702  1.10E-48  2.80E-44      43.56
                5                19-22S-regulator             cl_8              18         36         18         36          0         18       1825  8.80E-34  2.30E-29      28.64
                6                Pre-replication-complex      cl_17             14         16         14         16          0          2       1845  1.80E-33  4.70E-29      28.32
                7                Replication-complexes        cl_17             49         16         16         49         33          0       1812  3.60E-27  9.30E-23      22.03
                8                Replication-complex          cl_17             19         16         13         22          6          3       1839  3.00E-26  7.80E-22      21.11
                9                H+-transporting-ATPase-vacuolcl_9              15         34         14         35          1         20       1826  3.20E-25  8.20E-21      20.09
               10                20S-proteasome               cl_8              15         36         14         37          1         22       1824  8.60E-25  2.20E-20      19.65
               11                F0-F1-ATP-synthase           cl_9              15         34         13         36          2         21       1825  1.90E-22  5.00E-18       17.3
               12                Cytochrome-bc1-complex       cl_38              9          7          7          9          2          0       1852  2.40E-18  6.10E-14      13.21
               13                Oligosaccharyltransferase    cl_21              9         13          8         14          1          5       1847  3.30E-18  8.50E-14      13.07
               14                Replication-initiation-complecl_17              8         16          8         16          0          8       1845  3.70E-18  9.40E-14      13.03
               15                Replication-fork-complexes   cl_10             30         27         13         44         17         14       1817  4.30E-18  1.10E-13      12.95
               16                Anaphase-promoting-complex   cl_4              11         58         11         58          0         47       1803  1.00E-17  2.60E-13      12.58
               17                Cytochrome-c-oxidase         cl_27              8         10          7         11          1          3       1850  6.30E-17  1.60E-12      11.79
               18                RNA-polymerase-I             cl_2              14        149         14        149          0        135       1712  2.50E-16  6.40E-12      11.19
               19                Cdc28p-complexes             cl_4              10         58         10         58          0         48       1803  3.90E-16  1.00E-11         11
```

Figure 4 | Most significant associations between MCL clusters versus MIPS complexes. This figure shows only the top of the table returned by the program *compare-classes*. Each row represents the comparison between one complex (reference) and one MCL cluster (query).

31| Several alternatives are possible for the option ***Output type***, corresponding to various combinations of the query and reference graphs (union and difference). The arcs of the resulting graph will be labeled and colored differently depending on whether they belong to the query graph only, the reference graph only or to their intersection. For the study case, to only return the arcs that are in common to both graphs, select ***Intersection*** as ***Output type***.

32| If you want to visualize the resulting network with yEd or Cytoscape, select ***GML format*** as ***Output format***.

33| In case your graph is directed, check the option ***Graphs must be considered as directed***, so that an edge from node A to node B is considered as distinct from an edge from B to A. In our study case, protein interactions are undirected, so this option must remain unchecked.

34| Finally, you can indicate whether or not your graph may admit ***self-loops*** (edges having the same node as source and target). In our study case (synthetic lethality versus STRING interactions), the graph is undirected and has no self-loop.
▲ CRITICAL STEP The intersection statistics will be strongly affected by the nature of the graph (directed or not, with or without self-loops), as described in **Box 2**.

35| When this form is filled, click on the ***GO*** button. The computation of the comparison may take some time (between 10 s and a few minutes) depending on the size of the input networks.

36| The result page (**Fig. 5a**) shows statistics about the sizes of the input graphs, their union, intersection and differences (see **Box 2**) and a link pointing to a separate file corresponding to the comparison network. To save this network on your computer, right click on its URL and select ***Save link as***.... The resulting network can be visualized as described above (**Fig. 5b**).

37| The '***Next step***' box permits to use the network resulting from *compare-graphs* as input for some other NeAT programs (clustering, display, randomization, etc.).

Negative controls

38| To check that the results described previously were not obtained by chance only, we can run random negative controls by applying the processes described previously to random graphs. The program ***random-graph*** can be used to generate random graphs according to various random models. Click on the link ***Network randomization*** in the left menu. Upload a graph (e.g., *yeast_string_database_graph_names_undirected.tab*) and select the output format of your choice.
▲ CRITICAL STEP The most important parameter is the choice of the type of randomization. In general, we would recommend to select the option ***node degree conservation*** that consists in shuffling the edges, each node keeping the same number of neighbors as in the original graph. This procedure is specially designed to avoid duplicating edges, unless you check the option '***Allow duplicated edges***' (this should usually not be done). Another randomization type is the ***node degree distribution conservation*** where the global distribution of the node degree is conserved but each node presents a different degree than in the original graph. Finally, the program also supports ***Erdös-Renyi randomization***, where edges are distributed between pairs of nodes with equal probability.

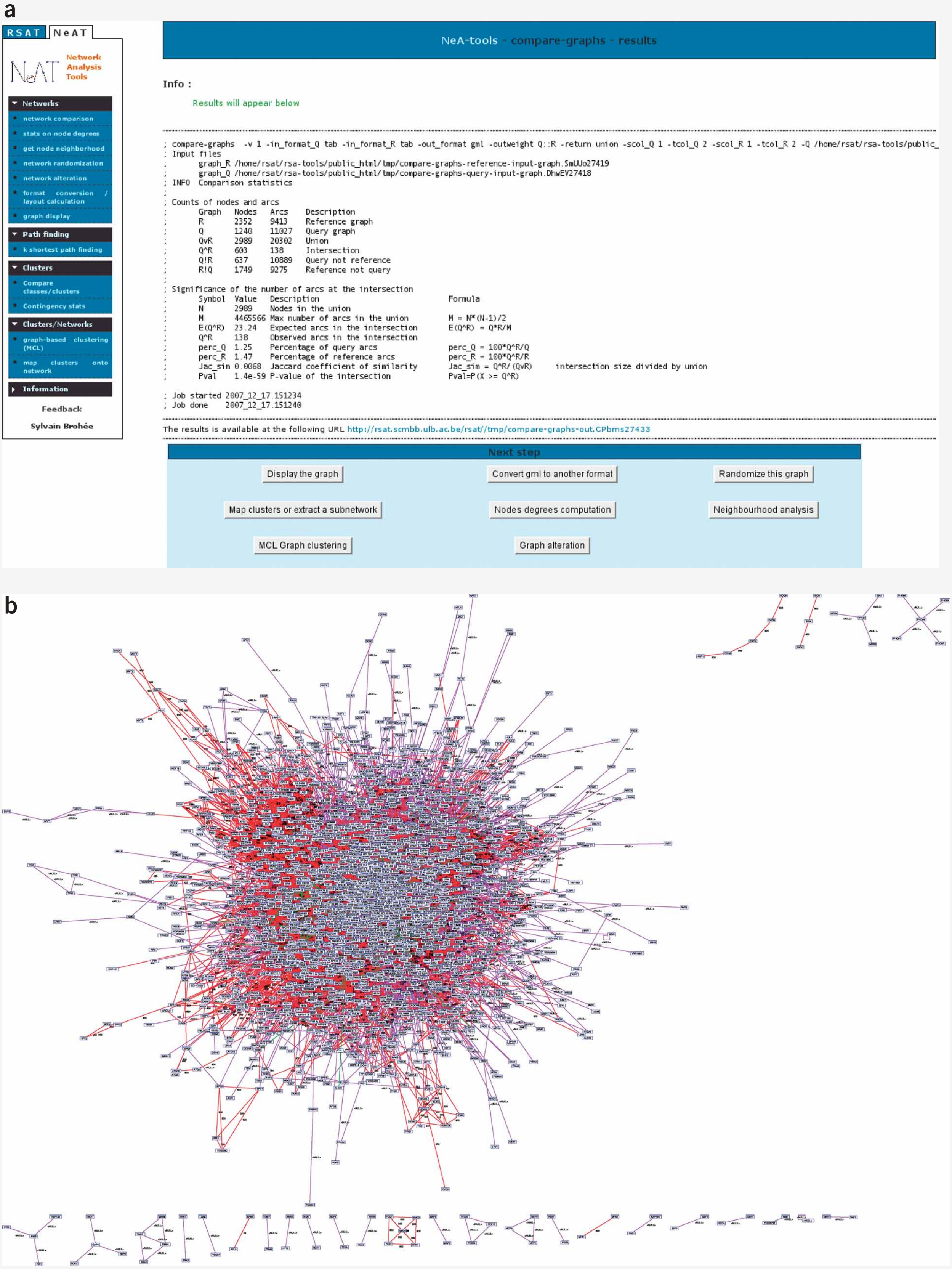
39| To obtain the randomized network, click on the ***GO*** button.

40| You can now apply to this randomized network all the steps described in the previous paragraphs (clustering, subnetwork extraction, comparison with reference graph, etc.). In principle, the results obtained with the randomized graph should be clearly less convincing than those obtained with the real STRING interaction network.

Path finding

41| Given an interaction network (e.g., the STRING database network) and two query proteins, we can ask which intermediate proteins connect them. This question can be answered using *Pathfinder*, a tool that retrieves the *k*-shortest paths in a network for given source and target nodes (see **Box 3** for more information on *k*-shortest paths finding). The STRING network with converted weights is available in the data repository (see EQUIPMENT), in the file *string_database_graph_converted_weights.tab*. Download this file to your computer.

42| In the NeAT main menu, click on the menu ***Path finding***, then on ***k shortest path finding*** to open the *Pathfinder* query form. Upload your network by clicking on the ***Browse...*** button in the section ***Network***. Alternatively, you can copy–paste the network into the text field. For the case you would like to store this network on the server for later use, click the



Info :

Results will appear below

```
; compare-graphs  -v 1 -in_format_Q tab -in_format_R tab -out_format gml -outweight Q::R -return union -scol_Q 1 -tcol_Q 2 -scol_R 1 -tcol_R 2 -Q /home/rsat/rsa-tools/public_
; Input files
;       graph_R /home/rsat/rsa-tools/public_html/tmp/compare-graphs-reference-input-graph.SmUUo27419
;       graph_Q /home/rsat/rsa-tools/public_html/tmp/compare-graphs-query-input-graph.DhwEV27418
; INFO  Comparison statistics
;
; Counts of nodes and arcs
;       Graph   Nodes   Arcs    Description
;       R       2352    9413    Reference graph
;       Q       1240    11027   Query graph
;       QvR     2989    20302   Union
;       Q^R     603     138     Intersection
;       Q!R     637     10889   Query not reference
;       R!Q     1749    9275    Reference not query
;
; Significance of the number of arcs at the intersection
;       Symbol  Value   Description                     Formula
;       N       2989    Nodes in the union
;       M       4465566 Max number of arcs in the union M = N*(N-1)/2
;       E(Q^R)  23.24   Expected arcs in the intersection       E(Q^R) = Q*R/M
;       Q^R     138     Observed arcs in the intersection
;       perc_Q  1.25    Percentage of query arcs        perc_Q = 100*Q^R/Q
;       perc_R  1.47    Percentage of reference arcs    perc_R = 100*Q^R/R
;       Jac_sim 0.0068  Jaccard coefficient of similarity       Jac_sim = Q^R/(QvR)     intersection size divided by union
;       Pval    1.4e-59 P-value of the intersection     Pval=P(X >= Q^R)

; Job started 2007_12_17.151234
; Job done    2007_12_17.151240
```

The results is available at the following URL http://rsat.scmbb.ulb.ac.be/rsat//tmp/compare-graphs-out.CPbms27433

Figure 5 | Result of the fusion between the BioGRID synthetic lethality data set (reference graph) and the yeast–protein interaction data set annotated in the STRING database (query graph). (**a**) Comparison statistics (see **Box 2**). (**b**) Drawing of the union graph (with yEd). Color code: red edges, false positives (edges found in the query graph but not in the reference graph); blue edges, false negatives (edges found in the reference graph but not in the query); green edges, true positives (edges present in both networks, in this case, only 138 among the ~20,000).

BOX 3 | *k*-SHORTEST PATHS FINDING IN WEIGHTED NETWORKS

Path finding attempts to find the shortest path between a given start node and a given end node in a network (graph). If several such paths exist, they should be all returned as equally valid solutions. Sometimes, we are not only interested in the shortest path, but also in the second shortest, third shortest or, in general, the *k*-shortest paths. The task of a *k*-shortest paths algorithm is to enumerate all paths up to the requested rank (*k*) in the order of their length. This is accomplished for example by the recursive enumeration algorithm[37] or by Eppstein's algorithm[38]. Often, the edges in biological networks are not equally relevant. For example, experimentally validated protein–protein interactions are more trustable than those observed in only one high-throughput experiment. To express such differential reliabilities, a higher *cost* (*weight*) can be placed on edges representing less trustable protein–protein interactions. When costs, or weights, have been set on the nodes or edges of a network, we no longer search for the *shortest* but for the *lightest* (that is less costly) path. Consequently, the *k*-shortest paths algorithm returns paths ranked according to their *weight* with the lightest path on top.

The weights have to be selected in a relevant way for the biological network of interest. The choice of a relevant weighting criterion clearly depends on your experience about this network and about the quality of the data available.

In a previous study[21,22], we evaluated the accuracy of *k*-shortest path finding for inferring metabolic pathways from compound/reaction networks, and showed that a graph where each node is weighted according to its degree (number of incoming + outgoing edges) achieves an accuracy of 83%.

In our study case with the yeast interaction network, we will use the scores provided by STRING as weights. In this case, the score assigned to an edge is a measurement for the reliability of the protein–protein interaction represented by this edge. In contrast, for Pathfinder, an edge weight is the cost of this edge. Therefore, we converted the scores into costs using the following formula:

$$W_e = \frac{1{,}000}{S_e},$$

where S_e is the score of an edge as defined in STRING (from 0 to 1,000), and W_e is the weight assigned to that edge for path finding.

Store network on server check box. This will allow you to perform further analyses on the same network, without having to transfer it repeatedly from your computer to the server.

43| Enter the IDs of the source and target nodes. For this study case, type RAS2 in the ***Source nodes*** field, and *TEC1* into the ***Target nodes*** field.

44| For the option ***Weighting scheme***, select '*as given in input graph*'. This will specify that weights should be taken from the third column of the input file and not calculated according to a predefined weighting scheme.

45| The result can be exported in various formats, depending on what you want to do with the resulting paths. (i) If you want to display the path in a tabular format, select ***Table*** as ***Output format***. (ii) The result can also be exported to GML format, to visualize the resulting paths as a subset of the original network (this can be done with visualization Cytoscape or yEd). For this, select ***Graph*** as ***Output format***, set the ***Graph format*** to ***GML format*** and set the ***Graph output type*** to ***paths unified into one graph***.

46| Click ***GO*** to start the computation.
? TROUBLESHOOTING

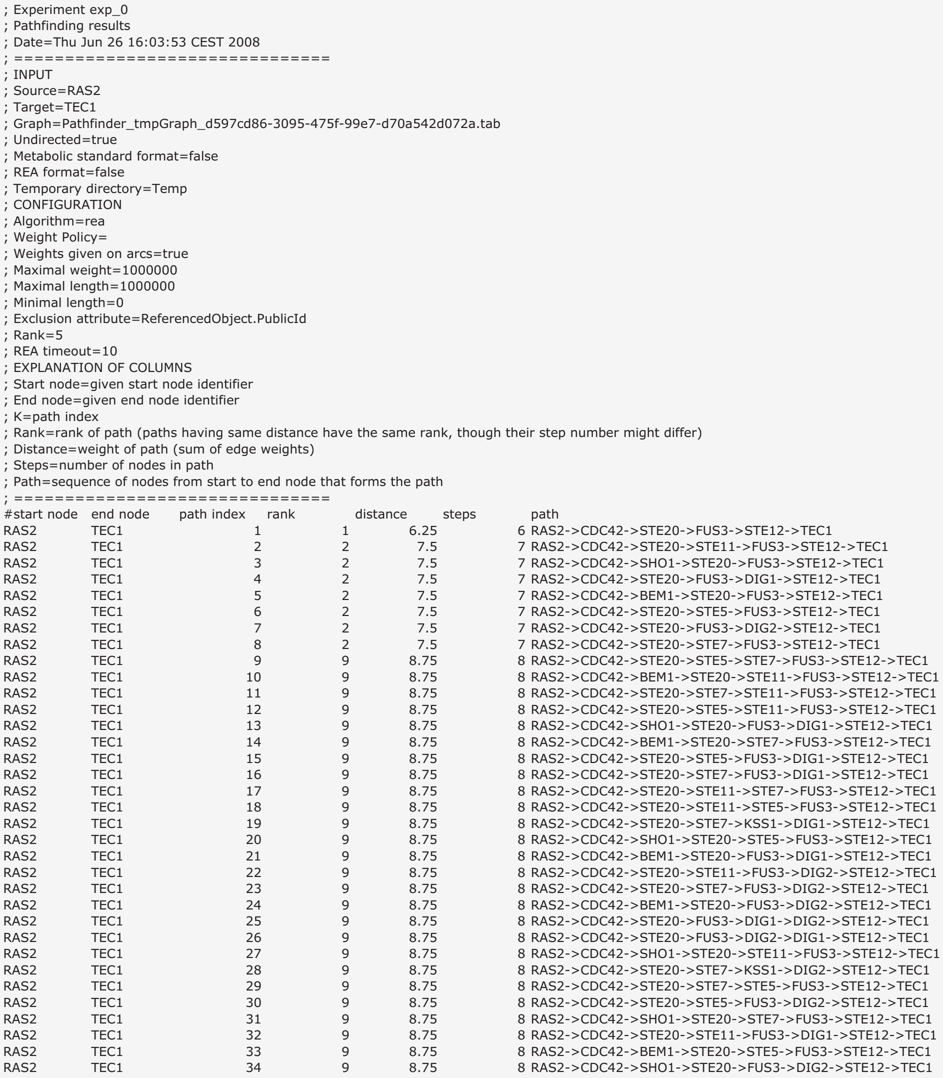
47| The result will be displayed according to the option chosen at Step 45. (i) Output format 'Table'. After a few seconds (or minutes, depending on the size of your graph), the results should appear in the form of two links. The first link points to the table of paths in simple text format, the second to the same table in HTML format (**Fig. 6**). If the checkbox ***Store network on server*** has been clicked, Pathfinder returns the identifier of the submitted network in addition. Submitting this identifier instead of the network itself speeds up the next path-finding job performed on it, because the previously transferred network is used, thereby avoiding to upload it again. (ii) The result form will contain a link to the resulting network in a GML file, which can be downloaded on your computer and displayed with Cytoscape or yEd. This link is followed by a '***Next steps***' box, which will permit you to fetch the result network into another NeAT tool.

● TIMING

The timings listed below depend on the server load (the number of jobs currently running on the server). However, for the study case we expect the tools to finish within 5 min.
Compare graph: <30 s; MCL: <20 s; graph-get-clusters: <40 s; display-graph: <1 min; compare-classes: <20 s; Pathfinder: <1 min; Fuzzy clustering: <2 min

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 1**.



```
; Experiment exp_0
; Pathfinding results
; Date=Thu Jun 26 16:03:53 CEST 2008
; ===============================
; INPUT
; Source=RAS2
; Target=TEC1
; Graph=Pathfinder_tmpGraph_d597cd86-3095-475f-99e7-d70a542d072a.tab
; Undirected=true
; Metabolic standard format=false
; REA format=false
; Temporary directory=Temp
; CONFIGURATION
; Algorithm=rea
; Weight Policy=
; Weights given on arcs=true
; Maximal weight=1000000
; Maximal length=1000000
; Minimal length=0
; Exclusion attribute=ReferencedObject.PublicId
; Rank=5
; REA timeout=10
; EXPLANATION OF COLUMNS
; Start node=given start node identifier
; End node=given end node identifier
; K=path index
; Rank=rank of path (paths having same distance have the same rank, though their step number might differ)
; Distance=weight of path (sum of edge weights)
; Steps=number of nodes in path
; Path=sequence of nodes from start to end node that forms the path
; ===============================
#start node  end node   path index   rank      distance     steps     path
RAS2         TEC1                1      1          6.25             6 RAS2->CDC42->STE20->FUS3->STE12->TEC1
RAS2         TEC1                2      2           7.5             7 RAS2->CDC42->STE20->STE11->FUS3->STE12->TEC1
RAS2         TEC1                3      2           7.5             7 RAS2->CDC42->SHO1->STE20->FUS3->STE12->TEC1
RAS2         TEC1                4      2           7.5             7 RAS2->CDC42->STE20->FUS3->DIG1->STE12->TEC1
RAS2         TEC1                5      2           7.5             7 RAS2->CDC42->BEM1->STE20->FUS3->STE12->TEC1
RAS2         TEC1                6      2           7.5             7 RAS2->CDC42->STE20->STE5->FUS3->STE12->TEC1
RAS2         TEC1                7      2           7.5             7 RAS2->CDC42->STE20->FUS3->DIG2->STE12->TEC1
RAS2         TEC1                8      2           7.5             7 RAS2->CDC42->STE20->STE7->FUS3->STE12->TEC1
RAS2         TEC1                9      9          8.75             8 RAS2->CDC42->STE20->STE5->STE7->FUS3->STE12->TEC1
RAS2         TEC1               10      9          8.75             8 RAS2->CDC42->BEM1->STE20->STE11->FUS3->STE12->TEC1
RAS2         TEC1               11      9          8.75             8 RAS2->CDC42->STE20->STE7->STE11->FUS3->STE12->TEC1
RAS2         TEC1               12      9          8.75             8 RAS2->CDC42->STE20->STE5->STE11->FUS3->STE12->TEC1
RAS2         TEC1               13      9          8.75             8 RAS2->CDC42->SHO1->STE20->FUS3->DIG1->STE12->TEC1
RAS2         TEC1               14      9          8.75             8 RAS2->CDC42->BEM1->STE20->STE7->FUS3->STE12->TEC1
RAS2         TEC1               15      9          8.75             8 RAS2->CDC42->STE20->STE5->FUS3->DIG1->STE12->TEC1
RAS2         TEC1               16      9          8.75             8 RAS2->CDC42->STE20->STE7->FUS3->DIG1->STE12->TEC1
RAS2         TEC1               17      9          8.75             8 RAS2->CDC42->STE20->STE11->STE7->FUS3->STE12->TEC1
RAS2         TEC1               18      9          8.75             8 RAS2->CDC42->STE20->STE11->STE5->FUS3->STE12->TEC1
RAS2         TEC1               19      9          8.75             8 RAS2->CDC42->STE20->STE7->KSS1->DIG1->STE12->TEC1
RAS2         TEC1               20      9          8.75             8 RAS2->CDC42->SHO1->STE20->STE5->FUS3->STE12->TEC1
RAS2         TEC1               21      9          8.75             8 RAS2->CDC42->BEM1->STE20->FUS3->DIG1->STE12->TEC1
RAS2         TEC1               22      9          8.75             8 RAS2->CDC42->STE20->STE11->FUS3->DIG2->STE12->TEC1
RAS2         TEC1               23      9          8.75             8 RAS2->CDC42->STE20->STE7->FUS3->DIG2->STE12->TEC1
RAS2         TEC1               24      9          8.75             8 RAS2->CDC42->BEM1->STE20->FUS3->DIG2->STE12->TEC1
RAS2         TEC1               25      9          8.75             8 RAS2->CDC42->STE20->FUS3->DIG1->DIG2->STE12->TEC1
RAS2         TEC1               26      9          8.75             8 RAS2->CDC42->STE20->FUS3->DIG2->DIG1->STE12->TEC1
RAS2         TEC1               27      9          8.75             8 RAS2->CDC42->SHO1->STE20->STE11->FUS3->STE12->TEC1
RAS2         TEC1               28      9          8.75             8 RAS2->CDC42->STE20->STE7->KSS1->DIG2->STE12->TEC1
RAS2         TEC1               29      9          8.75             8 RAS2->CDC42->STE20->STE7->STE5->FUS3->STE12->TEC1
RAS2         TEC1               30      9          8.75             8 RAS2->CDC42->STE20->STE5->FUS3->DIG2->STE12->TEC1
RAS2         TEC1               31      9          8.75             8 RAS2->CDC42->SHO1->STE20->STE7->FUS3->STE12->TEC1
RAS2         TEC1               32      9          8.75             8 RAS2->CDC42->STE20->STE11->FUS3->DIG1->STE12->TEC1
RAS2         TEC1               33      9          8.75             8 RAS2->CDC42->BEM1->STE20->STE5->FUS3->STE12->TEC1
RAS2         TEC1               34      9          8.75             8 RAS2->CDC42->SHO1->STE20->FUS3->DIG2->STE12->TEC1
```

Figure 6 | Result obtained with *Pathfinder* upon execution of protocol with the study case. The table lists the paths found between RAS2 (source node) and TEC1 (target node), ranked by increasing value of weight (distance). RAS2 and TEC1 are the start and end node of the filamentous growth pathway in yeast.

TABLE 1 | Troubleshooting table.

Step	Problem	Possible reason	Solution
8	After a few minutes, I still do not have any answer and the browser displays "Server is not responding"	If you submitted a heavy task, the processing may exceed 5 min. After that delay, Internet browser programs stop waiting for the server and display the error message Another possibility (if the task you submitted is not heavy) is that there is a problem with your Internet connection	For heavy tasks, it is preferable either to install the stand-alone version of the command-line tools on your machine or to write a client script for the RSAT Web services
17	No graph layout after having loaded a GML file into yEd or Cytoscape	When a graph is loaded in yEd or CytoScape, it is initially displayed with a trivial layout (all nodes on a diagonal)	In yEd: select **Layout** from the menu, then select the submenu **Organic** and choose the option **Classic**, then click **OK** In Cytoscape: select **Layout** from the menu, then select the submenu **yFiles** and choose **Organic** Both editors offer other layouts that you may try

(continued)

TABLE 1 | Troubleshooting table (continued).

Step	Problem	Possible reason	Solution
22	You obtain the message: "Error Incongruence between graph and cluster files"	Cluster and graph files do not correspond to the same network, or the specified format of the graph is not correct	Check that the clusters correspond to the graph. If so, check the format of your graph file is the one you entered in the relevant field
	The low-resolution heatmap does not display properly	The image is scaled so it fits on the window	Click on the image to zoom. You may inspect the whole map by using the scrolling bar
46	You obtain the message: "PATHFINDER ERROR: One of your seed nodes is not part of the input graph"	You provided seed node identifiers that do not match any of the node identifiers of the input graph	Check the spelling of your seed node identifiers In general, all tools require an exact match between input node identifiers and those of nodes in the network

ANTICIPATED RESULTS

Clustering

Figure 2 shows the results that should be obtained by applying the MCL graph clustering algorithm on the STRING database interaction network. Each cluster is highlighted with a specific color. **Figure 2a** only displays intracluster edges, so that each cluster appears as a separate component. This representation highlights the intracluster structure and edge density, and could give indication about possible improvement of the clusters by further subdivision. For example, the top-left cluster seems to be composed of several various connected regions, which could be explored in more detail, taking into account some biological knowledge. On **Figure 2b**, both intra- and interclusters are displayed. Intracluster edges are highlighted by cluster-specific colors, whereas intercluster edges are displayed in black, thereby revealing the interactions that were discarded during the clustering procedure. These two representations thus provide complementary indication for the interpretation of the clustering result.

The heatmap in **Figure 3** represents a section of the node–cluster membership matrix, with cells colored according to the membership degree (the darker the cell, the higher the membership value). Within each cell, the membership degree values are displayed, indicating how strongly each node (row) is connected to each cluster (column). This strength (node–cluster membership) is defined as the sum of weights of the edges connecting the considered node to the considered cluster, divided by the sum of weights of all edges starting from this node. The figure shows only a fragment of the table, but it already appears that some genes have similar membership profiles, thereby suggesting their involvement in common functions. This is the case of the genes *LPD1*, *PDA1* and *PYC1*, which are involved in pyruvate metabolism.

Figure 4 shows the results of the comparison between the dense clusters of the STRING graph and the complexes annotated in the MIPS database. The header gives a short explanation for the content of each column of the result table. In this case, results are sorted by decreasing values of the hypergeometric significance (last column) calculated as described in **Box 2**. Each row describes the comparison between one MCL cluster and one MIPS complex. For example, the first row compares the MCL cluster '*cl_2*', which contains 151 proteins, with a set of 138 proteins involved in cytoplasmic ribosomes. The intersection contains 123 proteins, which represents a very high fraction of both the MCL cluster, and the annotated complex. The probability to observe such an intersection by chance is 4E–146. The E value, obtained with the correction for multitesting, indicates that the number of false positives expected with such a *P* value would be 1E–141. In other terms, the correspondence between this MCL cluster and the cytoplasmic ribosome is too high to be explained by chance.

Network comparison

Figure 5a shows the statistics of comparison between the STRING 'database' network and the BioGRID synthetic lethality data set. The 'synthetic lethality' network used as reference contains 2,352 proteins linked by 9,413 edges, and the query graph contained 1,240 nodes and 11,027 edges. The intersection between those graphs is apparently weak: 138 edges only. The Jaccard coefficient indicates that this intersection represents no $>$0.68% of the union. However, the number of edges expected by chance at the intersection is even smaller: E(Q^R) $=$ 23.24. The hypergeometric *P* value (**Box 2**) indicates the probability to observe at least 138 edges at the intersection when 23.24 are expected by chance. In this study case, we observe that even with no $>$0.68% of edges at the intersection, the *P* value is very low (1.4E$-$59). In other terms, the number of edges at the intersection is too much high to be explained by chance, and is more likely to result from the biological relevance of both datasets.

Pathfinder

The known signal transduction path connecting RAS2 and TEC1 consists of the following steps[18]:
RAS2 - CDC42 - STE20 - STE11 - STE7 - KSS1 - DIG1/2 - TEC1

Pathfinder reports the following path of first rank (the matching parts are underlined, and the nonseed matching part are highlighted in bold):

RAS2 - **CDC42 - STE20** - FUS3 - STE12 - TEC1

This path connects STE20 to TEC1 via FUS3 and STE12, bypassing STE11, STE7, KSS1 and DIG1/2.

Among the paths of length 8 (third rank paths), we find paths closer to the annotated pathway, such as

RAS2 - **CDC42 - STE20 - STE11** - FUS3 - **DIG1** - STE12 - TEC1

Scott and colleagues applied their path-finding algorithm to another yeast protein–protein interaction network of similar size (4,500 nodes and 14,500 edges) taken from MIPS. For RAS2 and TEC1, they obtain the following as best path of length 8: RAS2 - CDC25 - HSP82 - **STE11** - STE5 - **STE7** - **KSS1** - TEC1

Although Pathfinder has not been designed in particular for protein interaction networks, it can be used to predict signal transduction pathways if appropriate weights have been set on the network under investigation. The accuracy of the prediction depends also on the data quality. For example, the STRING interaction network does not contain any edge between DIG2/DIG1 and TEC1, making it impossible to reach a prediction accuracy of 100%. When predicting pathways from real-world interaction networks, one must always keep in mind that these data might be incomplete or contain false positive interactions.

ACKNOWLEDGMENTS S.B. is the recipient of a PhD grant from the Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture (FRIA). K.F. is supported by the Actions de Recherches Concertées de la Communauté Française de Belgique (ARC grant number 04/09-307). G.L.-M. was funded by a PhD grant from the Fonds Xenophilia [Université Libre de Bruxelles (ULB)] and by a postdoctoral fellowship from the Région Wallonne de Belgique (TransMaze project 415925). The BiGRe laboratory is supported by the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265) and by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet). We acknowledge the students of the Master in Bioinformatics and Modeling (ULB, Belgium) for their useful corrections and suggestions. S.B. and K.E. equally contributed to this article.

Published online at http://www.natureprotocols.com/
Reprints and permissions information is available online at http://npg.nature.com/reprintsandpermissions/

1. Thomas-Chollier, M. *et al.* RSAT: regulatory sequence analysis tools. *Nucleic Acids Res.* **36**, W119–W127 (2008).
2. Brohée, S. *et al.* NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Res.* **36**, W444–W451 (2008).
3. Turatsinze, J.-V., Thomas-Chollier, M., Defrance, M. & van Helden, J. Using RSAT to scan genome sequences for transcription factor binding sites and *cis*-regulatory modules. *Nat. Protoc.* doi:10.1038/nprot.2008.97 (2008).
4. Defrance, M., Janky, R., Sand, O. & van Helden, J. Using RSAT oligo-analysis and dyad-analysis tools to discover regulatory signals in nucleic sequences. *Nat. Protoc.* doi:10.1038/nprot.2008.98 (2008).
5. Sand, O., Thomas-Chollier, M., Vervisch, E. & van Helden, J. Analyzing multiple data sets by interconnecting RSAT programs via SOAP Web services–an example with ChIP-chip data. *Nat. Protoc.* doi:10.1038/nprot.2008.99 (2008).
6. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N. & Barabási, A.L The large-scale organization of metabolic networks. *Nature* **407**, 651–654 (2000).
7. Jeong, H., Mason, S.P., Barabási, A.L. & Oltvai, Z.N. Lethality and centrality in protein networks. *Nature* **411**, 41–42 (2001).
8. Fell, D.A. & Wagner, A. The small world of metabolism. *Nat. Biotechnol.* **18**, 1121–1122 (2000).
9. Blatt, M., Wiseman, S. & Domany, E. Superparamagnetic clustering of data. *Phys. Rev. Lett.* **76**, 3251–3254 (1996).
10. Bader, G.D. & Hogue, C.W.V. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* **4**, 2 (2003).
11. Gagneur, J., Jackson, D.B. & Casari, G. Hierarchical analysis of dependency in metabolic networks. *Bioinformatics* **19**, 1027–1034 (2003).
12. Spirin, V. & Mirny, L.A. Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. USA* **100**, 12123–12128 (2003).
13. King, A.D., Przulj, N. & Jurisica, I. Protein complex prediction via cost-based clustering. *Bioinformatics* **20**, 3013–3020 (2004).
14. Van Dongen, S. *Graph Clustering by Flow Simulation*. PhD Thesis (Centers for Mathematics and Computer Science (CWI), University of Utrecht, 2000).
15. Pereira-Leal, J.B., Enright, A.J. & Ouzounis, C.A. Detection of functional modules from protein interaction networks. *Proteins* **54**, 49–57 (2004).
16. Enright, A.J., Van Dongen, S. & Ouzounis, C.A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30**, 1575–1584 (2002).
17. Brohée, S. & van Helden, J. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics* **7**, 488 (2006).
18. Scott, J., Ideker, T., Karp, R.M. & Sharan, R. Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.* **13**, 133–144 (2005).
19. Bebek, G. & Yang, J. PathFinder: mining signal transduction pathway segments from protein-protein interaction networks. *BMC Bioinformatics* **8**, 335 (2007).
20. Rahman, S.A., Advani, P., Schunk, R., Schrader, R. & Schomburg, D Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). *Bioinformatics* **21**, 1189–1193 (2004).
21. Croes, D., Couche, F., Wodak, S. & van Helden, J. Metabolic PathFinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Res.* **33**, W326–W330 (2005).
22. Croes, D., Couche, F., Wodak, S. & van Helden, J. Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.* **356**, 222–236 (2006).
23. Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).
24. de Nooy, W., Mrvar, A. & Batagelj, V. *Exploratory Social Network Analysis with Pajek* Series: Structural Analysis in the Social Sciences (No. 27) (Cambridge University Press, Cambridge, 2005).
25. Baitaluk, M., Sedova, M., Ray, A. & Gupta, A. BiologicalNetworks: visualization and analysis tool for systems biology. *Nucleic Acids Res.* **34**, W466–W471 (2006).
26. Hu, Z. *et al.* VisANT 3.0: new modules for pathway visualization, editing, prediction and construction. *Nucleic Acids Res.* **35**, W625–W632 (2007).
27. Hull, D. *et al.* Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.* **34**, W729–W732 (2006).
28. Lima-Mendez, G., van Helden, J., Toussaint, A. & Leplae, R. Reticulate representation of evolutionary and functional relationships between phage genomes. *Mol. Biol. Evol.* **25**, 762–777 (2008).
29. Croes, D., Couche, F., Wodak, S.J. & van Helden, J. Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.* **356**, 222–236 (2006).
30. Croes, D., Couche, F., Wodak, S.J. & van Helden, J. Metabolic PathFinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Res.* **33**, W326–W330 (2005).
31. von Mering, C. *et al.* STRING 7-recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res.* **35**, D358–D362 (2007).
32. Breitkreutz, B.J. *et al.* The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res.* **36**, D637–D640 (2008).
33. Keseler, I.M. *et al.* EcoCyc: a comprehensive database resource for *Escherichia coli*. *Nucleic Acids Res.* **33**, D334–D337 (2005).
34. Kanehisa, M., Goto, S., Kawashima, S. & Nakaya, A. The KEGG databases at GenomeNet. *Nucleic Acids Res.* **30**, 42–46 (2002).
35. Mewes, H.W. *et al.* MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.* **32**, D41–D44 (2004).
36. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
37. Jimenez, V.M. & Marzal, A. Computing the K shortest paths: a new algorithm and an experimental comparison. In *Proceeding of the 3rd International Workshop on Algorithm Engineering (WAE 1999)* Vol. **1668**, 15–29 (Springer-Verlag, London, 1999).
38. Eppstein, D. Finding the k shortest paths. *SIAM J. Comput.* **28**, 652–673 (1998).

7 Discussion

7.1 Summary

7.1.1 Path finding in RPAIR networks

A path finding tool was developed, which predicts metabolic pathways in KEGG LIGAND and KEGG RPAIR networks and which is integrated into NeAT.

The evaluation of path finding in RPAIR networks showed that RPAIR annotation, combined with a weight policy that penalizes hub compounds, yields a higher accuracy than either RPAIR annotation or weight policy alone. This is a step forward with respect to the previous results by Didier Croes, whose work was the starting point for this thesis.

7.1.2 Multiple-end pathway prediction by subgraph extraction

The evaluation of several multiple-end pathway prediction approaches showed that a combination of a random walk-based (kWalks) with a shortest-paths based (Takahashi-Matsuyama) approach reaches the highest overall accuracy. In addition, kWalks can discover weights. In the absence of a good weight policy, these weights increase pathway prediction accuracy considerably. This can be helpful when pathway prediction is applied to biological networks other than metabolic networks, for which a good weight policy is not yet known.

The subgraph extraction approaches are publicly available within the NeAT tool "Pathway extraction". This tool accepts compounds as well as reactions, reactant pairs, EC numbers, enzymes or genes as seeds. The user can submit custom groups of seeds or select a predefined seed node grouping strategy (for instance, if enzymes are provided as seeds, their associated reactions can be grouped according to their EC numbers). In addition, the tool offers networks from the major metabolic databases KEGG and MetaCyc and maps reference maps or pathways from these databases onto the predicted pathway. Last but not least, the tool can take custom networks and custom weights as input.

7.1.3 Application to microarray data set

Multiple-end pathway prediction was applied to a microarray data set that measured differential gene expression in the presence of each of 20 different compounds as sole nitrogen source with respect to urea as reference nitrogen source. Given the top five differentially expressed enzyme-coding genes for each condition, multiple-end pathway prediction suggested pathways that are up- or down-regulated in the presence of a specific nitrogen source.

For the good nitrogen source aspartate, which enables quick growth, the down-regulation of purine and proline degradation was predicted as well as the up-regulation of glycerol biosynthesis.

For leucine, which clusters with the bad nitrogen sources, but is stated to support quicker growth in [63], proline and purine degradation are predicted to be down-regulated. Proline degradation is not down-regulated for any other of the bad nitrogen sources, whereas purine degradation is also down-regulated in the presence of methionine and isoleucine. The pathway predicted for the up-regulated genes contains a part of the tyrosine degradation pathway.

For the intermediate nitrogen source phenylalanine, the purine and allantoin degradation pathways as well as the glyoxylate cycle are predicted to be down-regulated, whereas a part of the tyrosine/phenylalanine degradation pathway is predicted to be up-regulated.

In general, down-regulated pathways are very similar for good, intermediate and bad nitrogen sources and may rather be specific to urea (the reference nitrogen source) than the investigated nitrogen source. Up-regulated pathways are more variable and include storage compound synthesis as well as degradation pathways for the given nitrogen source in some cases.

Pathway prediction faced several problems mostly related to gene-reaction mapping. Results could sometimes not be obtained from a yeast-specific network constructed from KEGG PATHWAY, highlighting the importance of the mapping problem. The prediction results have therefore to be considered as preliminary.

7.1.4 Stoichiometric versus non-stoichiometric pathway prediction

The question can be raised whether or not metabolic pathways should be stoichiometrically balanced. Stoichiometric pathway prediction is recommended if the metabolic network of an organism (or a set of organisms) is well known. In cases where the metabolic network may be incomplete, non-stoichiometric approaches are more appropriate, since they are more robust with respect to missing reactions.

7.2 Strengths of pathway prediction

The multiple-end pathway prediction approaches developed during this thesis have several strengths, some of them unique (e.g. the treatment of seed sets and the acceptance of both reactions and compounds):

- Pathway prediction by subgraph extraction is a generic approach that can be applied to any biological network.
- It can handle large networks ($\sim$ 6,000 reactions).
- It does not require any input apart from the metabolic network, the weight policy and the seed nodes. If kWalks relevances are used as weights, acceptable prediction accuracies can be reached even without an appropriate weight policy. In particular, external

compounds do not need to be specified and it is not assumed that metabolism is at steady state.

- The prediction approach can discover unknown pathways consisting of known components.
- Pathway prediction can be fine-tuned to favor certain nodes. For instance, in a *generic metabolic network*, i.e. a network consisting of all reactions and compounds present in a metabolic database, reactions known to occur in certain organisms might receive a weight much lower than other reactions, to favor the extraction of organism-specific subgraphs. Similarly, results from a high-throughput experiment can be incorporated by assigning a node weight that represents a score obtained from the experiment (e.g. a function of the p-value of differential expression of the gene associated to the node).
- Sets of seed nodes can be specified to reflect AND/OR relationships between seed nodes. An AND relationship holds between seed node sets, whereas an inclusive OR relationship exists between the members of a seed node set.
- The web application allows to predict pathways from networks constructed from the two major metabolic databases KEGG and MetaCyc.
- Both, compounds and reactions can be provided as seeds. Thus, the web application supports compounds, reactions, reactant pairs, EC numbers or genes/enzymes as seed nodes and handles the required association of these seeds to reactions, reactant pairs and compounds.
- For metabolic networks from MetaCyc or KEGG, the web application maps the predicted pathway to known pathways from MetaCyc or KEGG respectively.

7.3 Limitations of pathway prediction

When using the pathway prediction approaches presented in this thesis, one should be aware of their limitations:

- Data quality.
  As for any other prediction method, prediction accuracy depends on the quality of the data. If reactions or compounds are absent in the source database, pathways containing them cannot be predicted. If associations of genes to reactions are imprecise, the predicted pathway will be inaccurate.
- Cycles and spirals.
  Cyclic and spiral-shaped pathways such as TCA cycle and fatty acid biosynthesis can only be partly predicted in most cases. Two-end path finding relies on the enumeration of the K-shortest paths. Paths by definition do not contain cycles, and closed paths are not enumerated. Pathways composed of several paths of equal weight may contain cycles, however.

Likewise, most multiple-end pathway prediction algorithms tested in this thesis search for the minimum weight tree connecting the seed nodes. The minimum weight tree would no longer be of minimal weight if it contained an additional edge to close a cycle. As in two-end path finding, a pathway may be cyclic if it contains several paths having the same weight between two seed nodes.

KWalks is a special case, because in contrast to the shortest-paths based algorithms, it does not seek to minimize the weight of a subgraph, but instead to maximize its relevance. With the kWalks algorithm, a subgraph is built by adding the most relevant of the remaining edges at each step and checking whether the pathway is connected. Thus, cycles may be obtained in some cases.

- Most parsimonious pathway assumption.
  Path finding makes the assumption that pathways are as short (parsimonious) as biochemically possible. It may be argued that this makes sense for the cell, since the synthesis of an enzyme is costly. In many cases, this assumption indeed results in accurate predictions. However, not all pathways are designed to synthesize a product or degrade a compound with the minimal number of enzymatic steps. For example, the TCA cycle has not been optimized to consist of the smallest possible number of enzymes but to produce energy and precursors for some metabolic pathways (e.g. amino acid biosynthesis). Pathway prediction with multiple seeds alleviates this problem, as it can take intermediate reaction steps into account.

- Central metabolism.
  From the evaluations performed during this thesis, it emerged that pathway prediction is particularly weak for central pathways such as glycolysis, which are highly interconnected and where many different alternatives exist. The criteria employed by the path finding approach presented in this work and by other non-stoichiometric approaches (e.g. [138, 18]) are currently not sufficient to distinguish between valid and invalid central pathways. Multiple-seed pathway prediction alleviates the problem by allowing the incorporation of more information. Still, a high proportion of a central pathway needs to be provided (as seed nodes) in order to correctly predict it.

- Generic compounds and stereoisomers
  KEGG and MetaCyc contain generic compounds, e.g. compounds such as "alcohol" or "an amino acid". Generic compounds serve as substrates and products of generic reactions (i.e. reactions carried out by broad-specificity enzymes) and organize compounds in a hierarchy. In this tree-shaped hierarchy, compounds with specific structures form leaves at the bottom, which are merged into generic parent compound classes with increasing broadness. For instance, tryptophan (leave) is an aromatic amino acid (generic compound), which is an amino acid (higher level generic compound). Stereoisomers introduce a new level of detail at the bottom of this compound hierarchy. For instance, D- and L-tryptophan, two leaves in the compound tree, are the children of the tryptophan compound class. Ideally, pathway prediction should navigate the compound hierarchy for each compound to find its most specific representative in the hierarchy, but this has not been implemented for the present prediction approach.

- Polymeric reactions and compounds
  The networks constructed during this work do not contain polymeric reactions, because they need an annotation effort to distinguish between substrate and product (which are often identical as in the case of KEGG reaction R01790 involving starch). Polymeric compounds such as DNA or RNA are also excluded to simplify the prediction task. Therefore, pathways containing polymeric reactions or compounds cannot be predicted.

- Stoichiometries.
  Pathways are not stoichiometrically balanced. The implications of this have been discussed in detail in chapter 5.

- Irreversible reactions.
  In general, the direction of a pathway can only be predicted if irreversible reactions are preserved during the construction of the metabolic graph. The generic metabolic networks offered by NeAT pathway prediction tools do not contain irreversible reactions. However, custom graphs containing irreversible reactions can be uploaded by the user. However, for multiple-seed pathway prediction, it is possible that a predicted pathway contains irreversible reactions of opposite directions.

- Reaction group treatment.
  If genes or EC numbers are associated to several reactions, it is not clear how many of them are contributed to the pathway. As explained in the Introduction (section 1.3.3), this ambiguity is due mainly to two reasons: On the one hand, enzymes may contain several catalytic sites and thus be associated to several EC numbers. On the other hand, one EC number may comprise several reactions. Such EC numbers are often assigned to *broad-specificity enzymes*, which are enzymes that are not specific to one substrate. For instance, EC number 1.1.1.1 describes the conversion of an alcohol to a ketone and is associated to the broad-specificity enzyme alcohol dehydrogenase. From the many reactions associated to this EC number, only a sub-set may be relevant for the pathway to be predicted. To deal with these ambiguities, the pathway prediction web tool allows to group reactions associated to a gene according to their EC numbers. Thus, an AND relationship exists between the reactions in different EC number groups and an OR relationship exists between the reactions within one EC number group. However, this grouping does not solve the problem entirely, as there are some cases where one EC number contributes more than one of its reactions to the pathway (e.g. histidine biosynthesis) or where only one of two EC numbers of an enzyme participates in the pathway (lysine biosynthesis). In section 9.1, postprocessing is discussed as a strategy to deal with this problem.

7.4 Alternative prediction approaches

As seen in the introduction, many different metabolic pathway prediction approaches exist.

The multiple-end pathway prediction approach presented in this thesis is a large-scale approach, which is not overly sensitive to noisy data (evaluation was performed in low-quality networks), which does not require any input other than the metabolic network, a weight policy and seed nodes, which is flexible (accepts compounds and reactions as seeds, can be applied to any network), but which on average does not reach high prediction accuracies (i.e. accuracies above 90%).

A few steps can be taken to increase the accuracy of multiple-end pathway prediction (see section 7.6), but it has inherent limitations: It does not take into account enzyme kinetics, does not explicitly trace atoms and does not consider stoichiometry.

In the following, each of these shortcomings is discussed in more detail.

7.4.1 Enzyme kinetics

Detailed models of metabolism (e.g. [118]) explicitly take into account the (often non-linear) kinetics of enzymes using various approximation techniques ([70]). Currently, kinetic data on a large number of enzymes are gathered for model organisms such as *S. cerevisiae*, which will allow to extend these models to the genome scale (personal communication J.J. Heijnen). At this level of description, the metabolic network of the organism needs to be well known, and the purpose of the model is the accurate prediction of the (wild type or mutant) organism's behavior in given conditions instead of the reconstruction of a metabolic network, the discovery of novel pathways or the interpretation of omics data. Thus, detailed models integrating enzyme kinetics answer different questions about metabolism than pathway prediction does.

7.4.2 Atom tracing

Path finding in the RPAIR graph considers atom flow indirectly, but does not explicitly trace atoms. Other tools (e.g. [6, 18, 130]) do trace atoms through the compounds of a pathway. This may result in higher prediction accuracies, as it ensures that atoms from the source indeed reach the target(s). There is however a drawback to it: It is not straightforward to search paths between reactions. Indeed, none of the tools relying on atom tracing accepts reactions as input (see tables 1.6 and 1.7). Given a reaction, it is not clear which of its products is of interest. The selection of a substrate-product pair with maximal structural similarity (or with maximal carbon atom transfer) does not help, because often a pair of hub compounds such as NADP/NADPH displays the highest structural similarity and transfers the largest number of carbon atoms. What is needed is information on the role of compound pairs in a reaction, which is provided by the RPAIR database. For instance, the pair NADP/NADPH is classified as "cofac". It is an open question whether carbon atom transfer rules (a maximal number of carbon atoms from the source should arrive at the target) can be combined with reactant pair roles and whether this would indeed increase the accuracy of path finding between reactions.

7.4.3 Stoichiometry

Stoichiometric versus non-stoichiometric pathway prediction was already discussed in chapter 5. It is conceivable to combine both approaches. For instance, first all EMs could be enumerated that contain one of the seeds of interest. Selected EMs could then form the input network to pathway prediction, which could extract a sub-network containing other compounds of interest. In this case, pathway prediction would act as a zoom to inspect a complex EM solution. Another way to combine both approaches would be to check whether a pathway predicted by subgraph extraction is a subgraph of an EM. In this case, EMs would serve as a filter to discard predicted pathways that do not ensure the net production of a compound of interest.

7.5 Top-down versus bottom-up pathway prediction

In the introduction, pathway prediction was defined as the enumeration of biochemically feasible pathways that connect a set of seeds. This may be referred to as a bottom-up approach: Given the seed set, a part of the metabolic network is extracted which represents the predicted pathway.

Another approach, which was mentioned in the introduction (section 1.6.2) starts from the whole metabolic network and divides it in smaller units that correspond to metabolic modules (or pathways). For pathway prediction, this top-down approach is less interesting, since reactions or compounds of interest are not taken into account. However, it allows to define pathways by defining the partition procedure. For instance, in [68], the network is partitioned by maximizing the number of intra-module links and minimizing the number of inter-module links. Many of the modules thus obtained were close to the reference maps listed in KEGG. Others (especially those from central metabolism) corresponded to a mixture of KEGG reference maps.

The decomposition algorithm introduced by [60] groups compounds in the order of their *specificity* (i.e. number of reactions they are involved in), starting from the most specific compounds that are linked to the lowest number of reactions and ending with the least specific compounds such as ATP and ADP. It outputs a tree consisting of a hierarchy of clusters that reflects the hierarchical organization of metabolic networks as suggested by Ravasz et al. (see Introduction, section 1.6.2). The least specific compounds (corresponding to the hub compounds) form clusters at the root of the tree, whereas the most specific compound clusters form its leaves. When applied to the metabolic network of *E. coli*, the computed clusters correspond well to the metabolic pathways as defined by the operons annotated in RegulonDB [61].

In [62] the metabolic network of *E. coli* is pruned iteratively by removing highly connected compounds. Parts of the pruned network corresponded to pathways annotated in EcoCyc.

In the context of metabolic pathway prediction, there are two interesting applications of network partitioning and pruning.

First, if the modules obtained from the partitioning are close enough to reference metabolic pathways, then network partitioning would be an interesting way to generate a set of (organism-specific) reference pathways from metabolic databases where reference pathways are not available.

Second, a pruned network can speed up pathway prediction, but with the important drawback that pathways containing the pruned compounds cannot be predicted, thus prediction accuracy will be lost.

7.6 Tuning pathway prediction

Several strategies can be adopted to increase the accuracy of pathway prediction without modifying the approach itself:

- Weight. A more sophisticated weight policy can be used, which integrates scores from omics data.
- Network quality. Pathways can be predicted from a higher-quality network, which does not contain imbalanced reactions or redundant compound entries (such as C06623 and C11915 in previous versions of KEGG, [134]). One such network can be obtained from BioMeta [122].
- Network size. Pathways can be predicted from a smaller network, possibly one that is specific to a set of organisms.
- Seed node number. The better the pathway of interest is covered by the seed nodes, the higher will be the prediction accuracy (see section 7.8 on the extend of pathway coverage by associated enzyme-coding genes).

However, as the evaluations presented in chapter 2 and 3 have shown, reasonable prediction accuracies can be achieved for huge low-quality networks, a simple weight policy and a small seed nod number.

In general, the prediction accuracy is highly dependent on the pathway shape and the coverage of the pathway by the seed nodes. Thus, the accuracy is pathway-specific, not organism-specific. In general, pathways are predicted with higher accuracy, if they are

- linear or branched instead of cyclic or spiral-shaped and
- located in the periphery of the metabolic network (i.e. not well interconnected with other pathways) rather than at its center (where pathways are highly interconnected).

7.7 Condition-specificity of metabolic pathways

Many pathways are only activated in certain conditions. Thus, if one wants to predict a pathway that is active in an organism in the given conditions, one needs either to integrate gene expression or similar data into the prediction approach or one needs to work with the condition-specific metabolic network (which is in turn obtained from gene expression or similar data).

One of the strengths of multiple-end pathway prediction is that it can easily integrate gene expression data, thus allowing to predict condition-specific pathways.

In a recent article, Shlomi et al. attempt the reconstruction of tissue-specific metabolism from the human metabolic network by combining flux balance analysis with gene expression data [151]. Multiple-end pathway prediction from seed genes offers an alternative to the reconstruction of tissue-specific metabolism from gene expression data.

7.8 Functional association of genes

Many applications of multiple-end pathway prediction involve the prediction of a metabolic pathway from a set of associated, enzyme-coding genes. This assumes that genes associated by co-regulation (in operons and regulons), co-occurrence in phylogenetic profiles or associated in other ways are also functionally related.

Accurate pathway prediction is highly dependent on the number of seed nodes and their position in the pathway. It is therefore important to consider to what extend seed reactions obtained from associated genes can cover metabolic pathways.

7.8.1 Co-regulation of genes

In [79], the extend of transcriptional co-regulation of *S. cerevisiae* enzymes belonging to the same KEGG map has been measured and a high correlation was found between co-regulation and co-occurrence in the same KEGG map. The same authors also investigated the position of transcriptionally regulated enzymes in metabolic pathways and found that gene regulation often enforces the linearity of a metabolic pathway.

These findings are confirmed by Seshasayee and co-authors, who state that for *E. coli*, "linear stretches of the metabolic network are tightly co-regulated" and that moreover a "substantial proportion (58%) of co-regulated enzyme pairs reside in the same operon" [149].

According to these studies, there is a good chance that seeds obtained from gene expression data cover a pathway sufficiently well to predict it with high accuracy.

7.8.2 Other types of gene association

Lee et al. investigated gene clustering in 5 eukaryotic genomes and found that genes from 30% to 98% of the KEGG maps (depending on the genome) cluster significantly on the chromosome [100]. Von Mering and coworkers combined several association types in *E. coli* (co-occurrence in phylogenetic profiles, conserved gene neighborhood and gene fusion) to obtain groups of associated enzyme-coding genes [167]. They found that these groups overlap significantly with the reference pathways in EcoCyc. They also noted that there is no one-to-one relationship between gene groups and reference pathways: There are pathways covered by more than one gene group and gene groups involved in more than one pathway.

These observations support the statement that multiple-end pathway prediction is more appropriate to interpret functionally related genes than simple pathway mapping, because in contrast to pathway mapping, it does not assume a one-tone correspondence between the gene group and a metabolic pathway.

8 Perspectives

8.1 Increase of pathway prediction accuracy

Several suggestions that might increase the prediction accuracy and which were not explored in this thesis are listed below.

- Thermodynamical constraints. Thermodynamical constraints on reaction directions filter out thermodynamically infeasible pathways. As explained in section 1.3.2, the direction of a reaction depends on its change in Gibbs free energy. Mavrovouniotis [111] developed a group contribution method to estimate the standard Gibbs free energy of the formation of a compound from its structure. He proposed to apply these estimated standard Gibbs free energies as constraints on reactions for which no information on reversibility was available from databases. However, the direction of a reaction in physiological conditions is not only dependent on the standard Gibbs free energy change, but also on compound concentrations and the temperature (see Introduction, section 1.3.2). Thus, thermodynamical constraints may be difficult to compute from compound structure alone.

- Combination with EMs. As described in the discussion (section 7.4), a network consisting of selected elementary modes may serve to filter out pathways that do not allow the net production of a compound of interest.

- Improved weight policy. Node weights could be obtained by a machine learning technique from the reference pathways. However, the risk of over-fitting is high in this case, since the number of reference pathways is far below the number of possible weight policies.

- Explicit tracing of atoms. In this thesis, reactant pair mappings from the RPAIR database were employed, but atoms were not explicitly traced. As mentioned in the discussion (section 7.4), relying only on atom tracing makes it hard to predict pathways from seed reactions. However, a mixed approach that combines atom tracing (as many substrate atoms as possible should reach the product) with RPAIR roles could improve the prediction accuracy for seed compounds as well as reactions.

8.2 Improvement of pathway prediction

In this section, improvements are suggested that do not affect the accuracy, but ease the interpretation of results or extend the applicability of the approach.

First, improvements concerning the subgraph extraction are listed.

- Subgraph enumeration. To allow the inspection of alternatives, not only the lightest subgraph should be returned, but subgraphs up to a requested number ordered according to their weight.

- Subgraph p-value. It is of interest to know how likely it is to obtain a particular subgraph score for a given seed node number. To answer this question, several authors compute the p-value of a subgraph ([77, 3]). These authors repetitively extract subgraphs for a given number of randomly selected seeds to obtain a distribution of subgraph scores. The p-value of an extracted subgraph given a certain seed node number is then computed from the seed number-specific distribution. Since the subgraph extraction approaches developed in this thesis are computationally expensive, it could be envisaged to precompute these distributions for several seed node numbers.

- Subgraph filtering. In two-end path finding, a number of filter options exist. Some of these filter options (no paths above a certain length or weight, absence of specific nodes) could be implemented for multiple-end pathway prediction as well. For instance, the predicted pathway could be filtered to remove all inter-seed paths above a certain length or weight.

- Subgraph layout. An improved graph layout would improve the "readability" of the predicted pathway. Currently, a general-purpose graph software [44] is used to layout predicted pathways. This software could be replaced by tools specialized on the layout of biological pathways (e.g. [102]).

Another series of improvements concerns the data sets.

- More metabolic databases. The pathway extraction tool currently offers networks from KEGG and MetaCyc. Integration of other metabolic databases, e.g. Reactome and UM-BBD would allow the prediction of pathways in more specialized metabolic networks.

- Transporters and compartments. Another extension would be the prediction of metabolic pathways from metabolic networks that integrate data on transporters. For instance, the transport of glucose across the cell membrane could be described by the reaction: `glucose_extern` → `glucose_cytoplasm`. Ideally, the different cell compartments should also be described by these extended metabolic networks. A metabolic network consisting of several modules would result, where each module describes the metabolism in one compartment and where the same compound may occur more than once in different modules (e.g. glucose_extern, glucose_cytoplasm). Modules are connected by transport reactions. For such networks, an additional constraint may be necessary to minimize the number of transport reactions in a predicted pathway.

- Integration of other biological networks. Networks could not only integrate data on transporters and compartments, but also on enzyme regulation and signal transduction. Such integrated networks are already available for some organisms (e.g. [46, 165]).

8.3 Evaluation of pathway prediction applied to biological data

This thesis focussed mainly on the development and evaluation of the multiple-seed pathway prediction approach. The next step is the application of pathway prediction to biological data sets, which poses however some new challenges.

- Integration of experiment-derived scores. Genome-scale experiments generate data on thousands of reactions or compounds. These data can be integrated into pathway prediction by modifying node weights. Several ways to compute node weights from experiment data were proposed ([77, 40]). Given these weights, it is unclear whether a weight policy is still needed and if so, how to best combine weights derived from experiments with weights computed with the weight policy.
- Positive and negative test sets. The evaluation of pathway prediction on omics data requires known cases of up- or down-regulated and unaffected pathways as positive and negative test sets. These will be difficult to obtain, thus the evaluation could partly be carried out on artificial data as in [139, 40].
- Comparative evaluation. Since many different approaches for metabolic pathway prediction exist (see Introduction section 1.10.3), a CASP-like protocol (e.g. [115]) could be developed that compares their performance on a number of selected test cases.
- Experimental validation. Finally, experiments have to be performed (e.g. by ^{13}C-tracing) to confirm or refute the predicted pathways.

8.4 Applications of pathway prediction

Pathway prediction approach was originally developed to interpret microarray data. The approach is generic enough to be applied to other high-throughput data sets as well, e.g. to interpret changed enzyme levels (e.g. relative protein abundances measured by iTRAQ, [153]) or compound levels (e.g. measured by gas chromatography/time-of-SSight mass spectrometry, [106]) between two conditions.

Metabolic pathway prediction could also be useful in metabolic reconstruction (see Introduction, section 1.10.1). It can propose a pathway given a set of interesting (e.g. co-expressed) enzyme-coding genes and can thus serve as an alternative to pathway mapping, especially if the latter resulted in a set of incomplete pathways. If applied to organisms with known or predicted operons but unknown metabolism, metabolic pathway prediction can reconstruct metabolic pathways from enzyme-coding operons.

Another interesting application would be the prediction of biodegradation pathways, especially if several intermediates are known. In contrast to other prediction tools (e.g. [50, 80]), the multiple-end pathway prediction approach accepts both compounds and reactions as input, thus it can incorporate knowledge on intermediates as well as on participating enzymes (e.g. if the genome of involved organisms is known).

An example for a possible application of pathway prediction is the following case: In [12], a novel experimental technique is presented that determines the presence/absence of thousands of annotated reactions in an organism of interest. This technique, termed *reactome array*, was applied to two organisms (*Pseudomonas putida* and *Streptomyces coelicolor*) and three microbial communities (geothermal pool on a volcanic island, surface seawater and deep-sea hypersaline anoxic lake). Pathway prediction could elucidate which metabolic pathways are differentially active in these organisms and communities.

9 Materials and methods

9.1 Graph algorithms

This section summarizes the graph algorithms that were applied in this thesis.

The computational complexity of the algorithms is given using the following notation: n is the number of nodes of the input network, m is the number of edges/arcs in the input network, s is the number of seeds [1] or seed node groups and K is the number of requested paths.

9.1.1 Enumeration of the K-shortest paths

The K-shortest paths problem is the problem of enumerating, in increasing length, the K-shortest paths between a source and a target node in a graph or digraph. In a weighted graph or digraph, the K lightest paths should be enumerated.

REA

In this thesis, the recursive enumeration algorithm (REA) developed by Jimenez and Marzal [83] has been selected as K-shortest paths algorithm. Other algorithms for this problem exist (e.g. [51]). REA has a computational complexity of $O(m + Kn\log(m/n))$ and outputs one path at a time in order of increasing weight. The original REA source code written in C (with modifications by Pierre Schaus and Jean-Noël Monette) is called within a Java wrapper.

REA does not return paths, but walks (i.e. nodes may be repeated). However, in practice REA is sufficiently quick that non-simple paths can be filtered out. Other constraints applied on the paths (e.g. maximal path length and weight, mutual exclusion of reaction directions, mutual exclusion of node sets in general, absence/presence of user-provided nodes) are likewise implemented by filtering the REA output.

Filtering REA output is effective and convenient, but poses a problem. Since REA lists walks instead of paths, it can enumerate an infinite number of walks in case the input graph contains a cycle. Thus, a time limit is required to stop REA in case the requested paths do not exist or do not pass the filter criteria. In addition, an upper limit on the output walk number prevents out-of-memory errors. Both limits may prevent the complete enumeration of correct paths. In practice, this problem is only relevant in large, unweighted networks. The user receives a warning when REA was stopped due to these limits.

[1] The set of *seed nodes* is a sub-set of the nodes in the network.

Source and target node sets

By default, K-shortest paths algorithms enumerate paths between one source and one target. Using a graph transformation suggested by Olivier Hubaut and described in [47], paths between a set of sources and a set of targets can be enumerated. The idea is to introduce pseudo-nodes, each of which is connected to one seed node set. Then, paths can be enumerated between the pseudo-nodes, which are afterwards removed from the output paths. Figure 9.1 illustrates this concept.

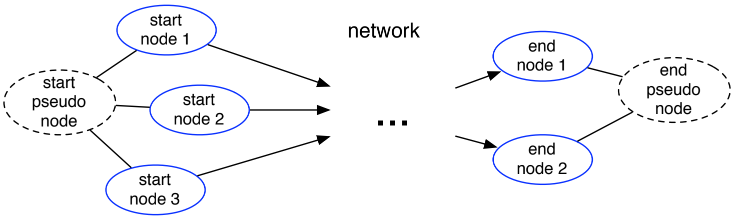


Figure 9.1: Pseudo-nodes are artificial nodes that are added to the input network to enable searches between seed node sets. For instance in K-shortest path finding, all start nodes are connected to a start pseudo-node and all end nodes to an end pseudo-node. Paths are enumerated between the start and the end pseudo-node, which are afterwards removed from the paths.

Paths and pathways

In the path finding approach presented in chapter 2, a pathway is considered to be the union of all first-ranked paths, i.e. all paths having the same weight. Thus, a pathway predicted by path finding may contain branches.

Directionality and symmetry

In a *symmetric graph*, the weight of the shortest path between any two nodes A, B $w(shortest\_path(A,B))$ equals $w(shortest\_path(B,A))$. Undirected graphs are obviously symmetric, since their adjacency matrix is symmetric. Importantly, the directed graphs evaluated in this thesis are also symmetric, because each reaction is represented by two reaction directions. This is exemplified in Figure 9.2. Thus, shortest paths between a node pair need to be computed in only one direction, either from A to B or from B to A. This symmetry no longer holds for graphs containing irreversible reactions. In these graphs, $shortest\_path(A,B)$ and $shortest\_path(B,A)$ both need to be computed to find the shortest paths between a node pair A and B.

9.1.2 Subgraph extraction from multiple seeds

Computationally, it is much more challenging to predict pathways from a seed set instead of two seeds. The enumeration of the lightest paths between two seeds or two seed sets is a

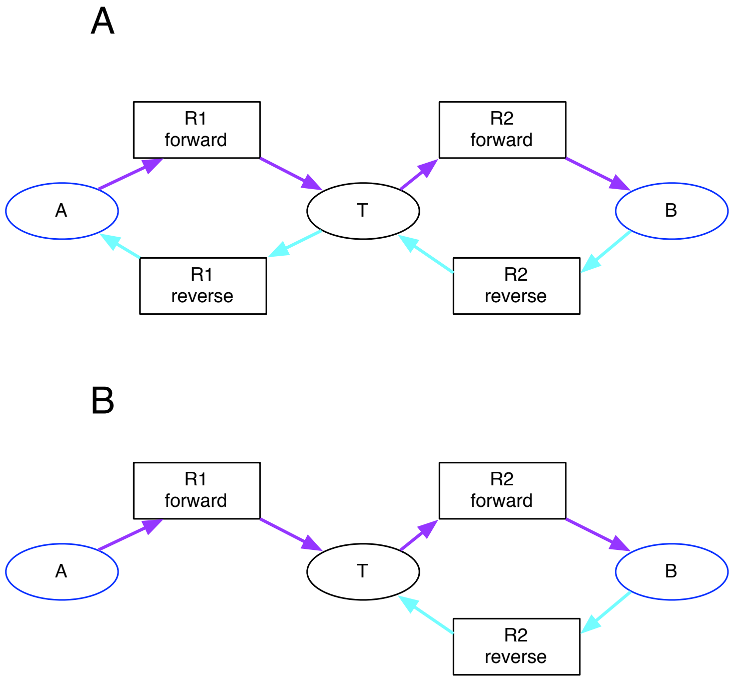


Figure 9.2: In a symmetric graph, the weight of the shortest path between nodes A and B is the same as the weight of the shortest path between B and A. This property holds also for directed metabolic graphs that include for each reaction both directions (Figure **A**). In this case, the shortest path between A and B (colored in violet) can be "mirrored" to obtain the shortest path between B and A (colored in cyan). Reactions R1 and R2 are reversible. However, if the metabolic graph contains irreversible reactions (e.g. R1 in **B**), this symmetry no longer holds (Figure **B**).

polynomial problem which can be solved optimally. In contrast, as will be discussed below, the extraction of the lightest subgraph is NP-hard, necessitating the use of heuristics.

During this thesis, seven subgraph extraction algorithms have been evaluated. They can be divided in random-walk based (kWalks), shortest-paths based (Klein-Ravi [95], Takahashi-Matsuyama [154], pairwise K-shortest paths) and hybrid algorithms (which combine each of the shortest-paths based algorithms with kWalks).

Given the weighted input graph, the shortest-paths based algorithms try to find a minimum-weight subgraph that connects the seed nodes. Thus, these algorithms try to solve the *Steiner tree problem*, which can be stated more formally as follows: Let S be a subset of the node set V in a weighted graph G. The Steiner tree problem is to find a minimum-weight tree subgraph of G that contains all the nodes in S (adapted from [67]). The minimum-weight tree subgraph is also referred to as lightest subgraph. The Steiner tree problem is NP-hard [89], thus it may be impossible to find an algorithm that solves it in polynomial time. The shortest-paths based algorithms use different heuristics to solve the Steiner tree problem. This means they are not guaranteed to find the lightest subgraph, but they find a subgraph that is at least close, if not identical to the lightest. Except for Klein-Ravi, which relies on the shortest paths algorithm of Dijkstra [39], the shortest-paths based algorithms find all shortest paths between two seeds by enumerating all first-ranked paths with REA.

KWalks takes an altogether different strategy: Instead of minimizing the weight of the subgraph connecting the seed nodes, it maximizes its relevance.

Although their strategies differ, all seven algorithms take the same input, namely:

- a weighted graph that can be either directed or undirected (except for Klein-Ravi, which only accepts undirected graphs)
- a set of seed node sets

From this input, they extract a subgraph, which represents the predicted pathway.

Multiple seed node sets

The graph transformation described in paragraph 9.1.1 can be extended to handle multiple seed sets (as proposed in [47]). In this case, each seed set is connected to one pseudo-node. A seed set is considered to be connected to the subgraph as soon as one of its members is connected to the subgraph. Thus, seed sets allow to express AND/OR relationships between seed nodes: The seeds within one set have an (inclusive) OR relationship, whereas seeds between sets have an AND relationship. When seeds are mentioned below, they may refer to the seeds in case one seed set is given or to the pseudo-nodes in case sets of seed sets are given.

Klein-Ravi

The algorithm by Klein and Ravi ([95]) is a heuristic to solve the node-weighted variant of the Steiner tree problem. First, the distance between any node pair in the graph is obtained with an all-to-all shortest paths algorithm (e.g. [39]). A set of trees is considered where each tree initially consists of a single seed node. At each step of the algorithm, a node and a subset of the remaining trees are selected such that the cost of tree merging is minimized. At least two trees have to be merged in each step. The cost of tree merging is computed as the sum of the weight of the selected node and the weights of the shortest paths between the selected node and the selected tree subset. This sum is divided by the number of trees in the selected subset. The algorithm terminates when all trees are merged. The computational complexity of this approach is $O(n^2 \log n + nm + ns^3 \log s)$.

The same implementation as in [147] was used, which was kindly provided by Nadja Betzler [15].

Takahashi-Matsuyama

The algorithm by Takahashi and Matsuyama ([154]) initializes the sub-network with a node chosen at random among the s seeds. It then proceeds by identifying in each step the lightest path(s) between any of the remaining seed nodes and any node in the sub-network. The lightest path(s) is merged with the sub-network. The computational complexity of this approach is $O(s(m + Kn \log(m/n)))$.

This algorithm has been implemented in Java, with the following modifications:

- Not the shortest, but the K-shortest paths are searched to ensure that all paths of the same weight are included in the solution subgraph. Thus, the subgraph extracted by this algorithm is not necessarily a tree.

- Instead of pre-computing the shortest paths between all node pairs as proposed by Takahashi and Matsuyama, only needed (K-shortest) paths are computed. This is done efficiently by introducing at each step a pair of pseudo-nodes, one of which is connected to all nodes in the current subgraph and the other is connected to all remaining seed nodes. This reduces the run-time from $O(s(n^2))$ as indicated by the authors to $O(s(m + Kn \log(m/n)))$, where $O(m + Kn \log(m/n)))$ is the run-time of REA.

- If no paths can be found between the initial single-node subgraph and the remaining seeds, the subgraph is initialized with another seed. This is repeated until either all seeds have served to initialize the subgraph or a single-node subgraph has been initialized that can be connected to one of the remaining seeds. This alleviates Takahashi-Matsuyama's weakness with respect to orphan seed treatment. Why this is important will be discussed in paragraph 9.1.2.

Pair-wise K-shortest paths

Pair-wise K-shortest paths has been developed during the initial stages of this thesis and can be considered as a heuristic to solve the Steiner tree problem. In the first step, REA is called successively on each pair of seed nodes. The resulting path sets are stored in a path matrix, and the minimal weight between each node pair is stored in a distance matrix. In the second step, the sub-network is constructed from the path sets, starting with the lightest path set. Step-wise, path sets are merged with the subgraph by increasing order of their weight. The process stops

Table 9.1: KWalks parameters.

Parameter	Default value	Explanation	Modified in the evaluation
Limited	true	Random walks are not infinite, but of limited length.	no
L	50	Maximal length of a random walk.	no
Up to	true	Random walks of less than the maximal step number are allowed.	no
Initial seed node probabilities	uniform distribution	Distribution of seed node probabilities.	no
Iteration number	1	The number of times kWalks is executed on a graph.	yes

if either all seeds belong to one connected component of the sub-network or all path sets have been merged with the sub-network.

The computational complexity of this approach is $O(s^2(m + Kn\log(m/n))$, because the REA algorithm is called s^2 times.

kWalks

The kWalks method is a generic algorithm ([48, 21]) to build a most relevant subgraph connecting seed nodes in a large graph. The relevance of an edge is measured as the expected number of times it is visited by random walks connecting seed nodes. These expected passage times reflect both the topology of the network and the edge weights.

A subgraph is obtained from the edge relevances by keeping only those edges above a minimal relevance threshold. This threshold is automatically fixed such that the subgraph induced by the selected edges is weakly connected. The sub-networks extracted by kWalks may contain branches ending in non-seed nodes. These branches are removed in a final pruning step.

KWalks takes some additional parameters not shared with the other subgraph extraction algorithms. An important one is the *iteration number*. The edge relevances computed by kWalks can serve as new edge weights. kWalks can then be run on the input graph with updated weights. This iterative process may be repeated a number of times to increase the difference between more and less relevant edges.

Table 9.1 lists the kWalks-specific parameters. The unlimited kWalks computes the edge and node relevances as the expected number of times edges and nodes are visited by walks of unlimited length. The limited kWalks is a computationally more efficient approximation of the unlimited case, where walk length is limited. Most of the parameters were not varied during evaluation, because they have reasonable default values. For instance, the parameter L (maximal length of random walks) is set to 50, because measurements on artificial networks have shown that very good precision/recall curves are obtained for this length [48].

In this thesis, the implementation of kWalks by Jerôme Callut was used, which was kindly provided by Pierre Dupont. The computational complexity of the unlimited kWalks is $O(sn^3)$ and of the limited kWalks is $O(sLm)$, where L is the maximal allowed length of the random walks. Since L is typically a small constant, kWalk's runtime increases linearly with the seed node number for a given input graph.

Hybrid subgraph extraction algorithms

As discussed in chapter 3, kWalks is more sensitive whereas the shortest-paths based algorithms are more specific with respect to pathway prediction accuracy. Therefore, they were combined in a hybrid approach.

Such a hybrid approach runs in two steps: kWalks extracts a sub-network representing a fixed proportion of the input network and the shortest-path based algorithm is launched on this intermediate sub-network to obtain the final pathway.

Combining kWalks with path-based approaches gives two new parameters:

- *Size of the sub-network* kWalks extracts a sub-network whose size is fixed to a given percentage of the number of nodes in the input network.
- *Input or computed weights* The path-based algorithms may either use the input weights or the edge/node relevances computed by kWalks.

The difficulties of assembling a metabolic pathway from multiple paths

The shortest-paths based algorithms construct a pathway from several paths. For metabolic graphs, this creates several problems, which have not yet been solved satisfactorily.

- Treatment of irreversible reactions. In graphs containing irreversible reactions, it is not ensured that the irreversible reactions in the subgraph operate in the same direction.
- Mutual exclusion of reactant pairs across paths. This is currently only implemented for Takahashi-Matsuyama, at the cost of rendering the resulting subgraph seed-node-order dependent. Thus, this algorithm could return different subgraphs when repetitively executed on the same seeds and input RPAIR graph. However, such behavior is rarely observed in practice.

Orphan seed nodes and disconnected components

A predicted metabolic pathway may contain orphan seeds or may consist of several components.

Orphan seeds are seed nodes or seed node sets that could not be connected to the subgraph. Orphan seeds can occur for the following reasons:

- No path exists between the orphan and the subgraph in the input graph, either because the input graph consists of several components or contains irreversible reactions.

- A path exists, but does not pass the filter criteria (e.g. it contains nodes already excluded, or does not contain nodes required to be present).

A pathway may connect all seeds and seed node groups, but may not be a connected graph. This can occur when several seed node sets have been given as input, as depicted in Figure 9.3A.

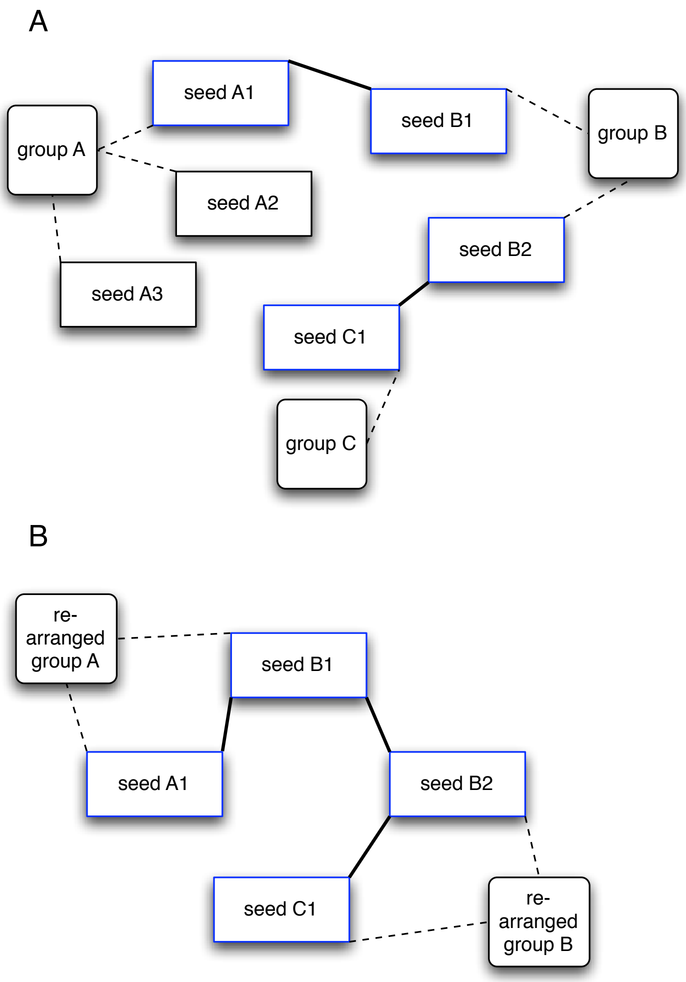


Figure 9.3: A subgraph is extracted for the three seed node sets group A, group B and group C. Group A is connected to group B via seed B1, but group C is connected to group B via seed B2. Thus, two components are formed, one containing seed A1 and B1 and the other containing seed C1 and B2 (Figure **A**). After re-grouping of the seed sets, it is possible that the two components are connected (in this case by an intra-group connection with respect to the original seed groups) (Figure **B**).

Preprocessing and postprocessing

Preprocessing

Preprocessing eases the task of subgraph extraction by connecting seeds that can be treated without heavy computation. For instance, neighboring seed nodes such as a reaction and its substrate can be directly connected. The treatment of indirect neighbors (those that are separated by one node, e.g. substrate and product of a reaction) is less obvious. In a weighted graph, a path with many intermediate nodes may be lighter than a path with only one intermediate node. However, in the RPAIR network, additional information on the intermediate node becomes available. Consider for instance two seed reactant pairs that share a main compound. It makes sense biochemically to link them via this compound. It is still possible that a lighter path exist, but it is less likely that this path is biochemically more relevant. A small-scale evaluation on four branched pathways manually mapped from MetaCyc onto KEGG showed that preprocessing enhances both accuracy and speed of computation. In the absence of a thorough evaluation, the decision on whether or not this option should be enabled is left to the user.

Postprocessing

As discussed in section 9.1.2, the extracted subgraph may consist of several components. In some situations, it may make sense to repeat pathway prediction with re-grouped seed sets.

Consider for instance an enzyme B that contributes two reactions B1 and B2 to a pathway. This enzyme will be treated as a seed node group consisting of two seeds. During conventional pathway prediction, at least one of these seed reactions will be connected to the subgraph (assuming the seeds are not orphans), but the second reaction is not necessarily connected. If pathway prediction is repeated with seed sets re-arranged such that the seed nodes of each component form a new seed set (see Figure 9.3B), the second reaction of the enzyme may be connected to the pathway. However, the solution obtained after postprocessing is not necessarily identical to the solution that would have been obtained had the seeds been given separately.

Treating all reactions associated to an enzyme as separate seeds is not a satisfying solution either, because due to the imprecise enzyme-reaction mappings not all of them may be relevant for the pathway. Thus, seed sets in combination with postprocessing might be a way to deal with these imprecise mappings. However, without an evaluation, it is not clear whether postprocessing is really helpful in these cases and is therefore left as an option to the user.

Comparison of the subgraph extraction algorithms

Differential prediction accuracy of subgraph extraction algorithms

Chapter 3 presented a comparison of the subgraph extraction algorithms with respect to their pathway prediction accuracy. This section discusses likely reasons for the different accuracies reached by the subgraph extraction algorithms. The performance of the algorithms might differ for other networks, for instance the KEGG RPAIR network. Unfortunately, the accuracy of the algorithms could not be evaluated for the KEGG RPAIR network, because of the absence of a sufficient number of branched reference pathways with KEGG identifiers. KEGG maps are not suitable for reasons discussed in the introduction, section 1.4. The aMAZE pathways are not an appropriate reference pathway set either, because of their small number of branched pathways. Careful mapping of four selected MetaCyc pathways onto KEGG showed that automatic mapping without curation would be inaccurate and it was beyond the scope of this thesis to manually map all the MetaCyc pathways onto KEGG. However, it would be highly interesting to compile such a reference pathway set in the future in order to evaluate systematically the performance of the subgraph extraction algorithms for KEGG RPAIR networks.

In the compound-weighted MetaCyc network, kWalks, even if iterated, does not reach the accuracy of shortest-paths based approaches. One possible reason may be that the current version of kWalks does not handle mutual exclusion between forward and reverse direction of reactions. Thus walks can cross reactions from substrate to product to another substrate, which results in an irrelevant pathway.

Klein-Ravi has the lowest prediction accuracy of the three shortest-paths based algorithms, very likely because in its current implementation it does not accept directed networks. Its accuracy may be higher for the RPAIR network, which can be undirected for reasons discussed in chapter 2. Takahashi-Matsuyama outperforms pairwise K-shortest paths by 8%. The reason is that pairwise K-shortest paths only searches shortest paths between seed nodes, whereas Takahashi-Matsuyama also takes into account shortest paths between seed nodes and other subgraph nodes. Thus, K-shortest paths can miss a path between a seed node and a subgraph node that is lighter than any path to another seed node, which would be detected by Takahashi-Matsuyama.

All three shortest-paths based approaches yield a small yet significant increase in accuracy when combined with kWalks (as measured with a paired signed Wilcoxon rank test). Apparently, kWalks discards nodes that would otherwise be predicted as false positives in the pathway. However, kWalks' main role is to increase the computational speed of pathway prediction.

Strengths and weaknesses of subgraph algorithms

Apart from their prediction accuracy, the algorithms have other strengths and weaknesses that are discussed here.

To summarize: There is no single subgraph extraction algorithm that can be recommended for all situations. In most situations however, the winner of the evaluation described in chapter 3, namely the hybrid of Takahashi-Matsuyama and kWalks, should be used.

In general, hybrids are quicker and more accurate than their shortest-paths based pendants, but they have a shortcoming: Whereas the repetition of the shortest-paths based algorithms does not change the solution (except for Takahashi-Matsuyama as discussed in 9.1.2), it was observed that different solutions were returned when executing one of the random-walk based algorithms repetitively on the same seeds. This behavior may probably be due to ties in the edges relevances. From two edges of equal relevance, one is selected at random and added to the growing subgraph. When the selected edge is the last needed to meet the threshold criterion, the algorithm stops. Which of the two edges is incorporated into the subgraph can therefore differ upon repetition of the subgraph extraction. However, in most cases only one or a very small set of solutions is returned for repetitive executions.

KWalks is the quickest of all the tested subgraph extraction algorithms and in addition can discover weights. In the absence of a weight policy, extracting a subgraph with the edge relevances returned by kWalks instead of unit weights increases the accuracy of the pairwise K-shortest paths algorithm by more than 10%. Moreover, kWalks allows to refine the relationship between seed nodes by assigning probabilities to them. It is also the only one of the tested algorithms that can extract other than trivial cyclic pathways.

However, kWalks has to be iterated to reach a high accuracy, which decreases its speed considerably. Even iterated kWalks was not among the top five algorithms with highest accuracy, for reasons discussed in section 9.1.2.

Klein-Ravi is by far the quickest among the three shortest-paths based algorithms, but neither does it take into account several paths of equal weight nor is it applicable to directed networks.

Pairwise K-shortest paths is the slowest of the three shortest-paths based algorithms, because it computes the K-shortest paths between each pair of seed nodes, thus its runtime increases quadratically with the seed node number instead of linearly as for Takahashi-Matsuyama.

However, pairwise K-shortest paths is robust with respect to orphan seed nodes, whereas Takahashi-Matsuyama is not. At each step, Takahashi-Matusyama computes the K-shortest paths between all remaining seed nodes and the subgraph. As soon as Takahashi-Matsuyama cannot connect any of the remaining seed nodes to the subgraph, it stops and returns an unconnected subgraph. However, even though it is not possible to connect the seed nodes to the subgraph, it may be possible to connect them among each other, thus obtaining a subgraph with several components. In order to connect non-orphan seeds among each other, Takahashi-Matsuyama would have to test all $s \times (s-1)$ seed node combinations and thus would reach the runtime of pairwise K-shortest paths.

Simply using the largest connected component of a network does not avoid orphans, since in RPAIR networks, reactant pairs exclude each other mutually. Thus, whether a seed node is an orphan or not depends on the other seed nodes. Even worse, the order of seed nodes in RPAIR networks matters, because the nodes that are forbidden (by mutual exclusion) depend on which seed node is selected first. As mentioned in 9.1.2 reactant pair exclusion across paths is only implemented for the Takahashi-Matsuyama algorithm.

Apart from the better treatment of orphans, another advantage of pairwise K-shortest paths is that it outputs a distance matrix for the seeds, which allows to group them by standard cluster algorithms. Pairwise K-shortest paths comes with an option to generate a dendrogram of the seed nodes. Such a dendrogram cannot be obtained from Takahashi-Matsuyama.

By default, Takahashi-Matsuyama should be favored over pairwise K-shortest paths, because it is quicker, more accurate and takes mutual exclusion of reactant pairs across paths into account. In special situations and if the graph is not an RPAIR graph, pairwise K-shortest paths can be run to connect orphans among each other or to group seeds using the seed distance matrix.

Exact solution of the Steiner tree problem

In [105], an algorithm solving the Steiner tree problem exactly is described. This algorithm has not been used because it requires commercial software to be installed (CPLEX) and because it is unclear whether it could be customized to treat mutual exclusion of nodes (required for

Table 9.2: Composition of selected metabolic networks that were employed for the evaluation of pathway prediction.

Network	Database (version)	Directed*	Compound number	Reaction number	Arc/edge number	Evaluation
reaction graph	KEGG LIGAND (41.0)	true	5,312	6,359 × 2	53,572	path finding
RPAIR graph	KEGG RPAIR (41.0)	false	4,297	7,058	28,232	path finding
reaction-specific RPAIR graph	KEGG LIGAND/ RPAIR (41.0)	false	4,297	12,828	51,312	path finding
MetaCyc graph	MetaCyc (11.0)	true	4,891	5,358 × 2	43,938	pathway prediction with multiple seeds

* In a directed graph, reactions are duplicated to represent both directions. Consequently, the arc number between a reaction and a compound is also duplicated.

treatment of reaction directions and reactant pairs). If these obstacles could be overcome, it would be worth evaluating this algorithm.

9.2 Networks and reference pathways

9.2.1 Metabolic networks

During this thesis, pathway prediction was evaluated with networks constructed from both KEGG and MetaCyc. KEGG networks were built in a variety of ways: with selected compounds or reactant pair classes removed and as directed or undirected graphs. An exhaustive list of all KEGG networks constructed is available at http://rsat.ulb.ac.be/pathfindingsupplementref/MetabolicGraphs.html. Likewise, several versions of the MetaCyc network were built (details can be found in chapter 3).

Table 9.2 lists properties for a selection of these networks, namely all those that are supported by NeAT (of course, since the networks in NeAT are updated, their version and consequently their node and edge numbers may be different from those indicated in the table).

9.2.2 Metabolic reference pathways

In general, reference pathways were modified as follows: Terminal compound nodes were removed from all reference pathways, because the evaluation took place between seed reactions.

Table 9.3: Filtering steps applied to the aMAZE reference pathways.

All organisms	*E. coli*	*S. cerevisiae*	*H.sapiens*
Pathways before filtering			
Pathways (116)	55	29	32
Cyclic* pathways (7)	2	4	1
Branched* pathways (25)	13	5	7
Pathways with less than 3 reactions (46)	20	14	12
Pathways after filtering and linearization			
Pathways (69)	37	14	18
Pathways after mapping of reactions to reactant pairs			
Pathways (55)	32	11	12

* BioPool compounds not counted

After removal of terminal compounds, only reference pathways with at least five nodes were kept, to avoid trivial predictions. A number of more specific filter steps and modifications was applied to each reference pathway set separately.

Reference pathways for path finding

Path finding was evaluated with reference pathways from three organisms taken from aMAZE. Table 9.3 summarizes the filtering steps, which reduced the 116 aMAZE pathways to the 55 linearized reference pathways used for evaluation. Figure 9.4 displays a histogram of the node numbers for the final pathway set. Since the pathways are linearized, the node number of a pathway indicate its length.

The linearized aMAZE pathways can be obtained in various formats from `http://rsat.ulb.ac.be/pathfindingsupplementref/ReferencePathways.html`. The unmodified aMAZE pathways are available as part of the metabolic database described in section 9.3.

Reference pathways for multiple seed pathway prediction

As can be seen from Table 9.3, only 25 of the aMAZE pathways are branched. To evaluate pathway prediction with multiple seeds on a larger set of branched pathways, the 171 reference pathways annotated for *S. cerevisiae* in MetaCyc (version 11.0) were parsed from the organism-specific biopax.owl and pathways.dat files. The latter file contains the pathway-specific classification of compounds into side and main, which is not included in the OWL file. From these reference pathways, a large number had to be removed for the following reasons:

- They involved non-small molecules.
- They consisted of several components.

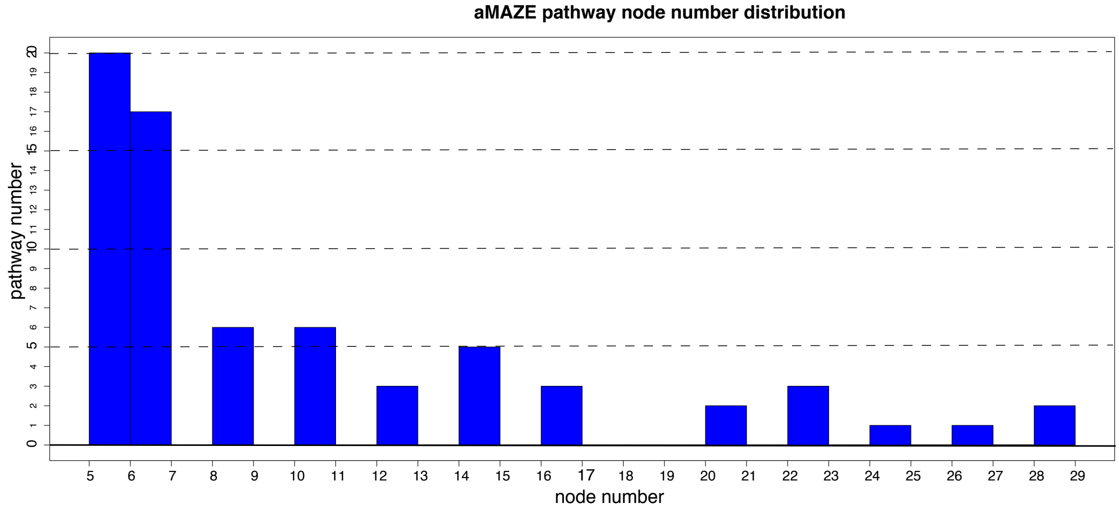


Figure 9.4: Histogram of pathway node numbers for the 55 reference aMAZE pathways used for the evaluation of two-end pathway prediction. Since the pathways are linearized, this plot also reflects their length distribution. All pathways below five nodes have been discarded.

- They contained BioCyc identifiers that were absent from MetaCyc.
- They consisted of less than five nodes.

From the 78 remaining pathways, seven more were removed because they contained nodes absent from the largest connected component of MetaCyc. Finally, 71 pathways remained for evaluation, more than half of them branched and/or cyclic. Thus, the MetaCyc reference pathway set poses a real challenge for pathway prediction. Figure 9.5 plots the distribution of node numbers for the final pathway set.

Table 9.4 lists the MetaCyc pathways that were kept for evaluation along with their properties.

Table 9.4: The 71 reference *S. cerevisiae* pathways obtained from MetaCyc (11.0) that were used for the evaluation of pathway prediction with multiple seeds.

Pathway name	Node number	Arc number	Number of branches	Cycles present
4-hydroxyproline degradation	7	6	0	false
aldoxime degradation	5	4	0	false
allantoin degradation	12	11	2	false
arginine biosynthesis III	12	13	4	true
asparagine degradation I	5	4	0	false
aspartate superpathway 1	7	6	0	false
aspartate superpathway 2	17	16	4	false
aspartate superpathway 3	10	9	2	false

bifidum pathway	29	33	12	true
butanediol fermentation	36	47	18	true
chorismate biosynthesis	13	12	0	false
cysteine biosynthesis II	9	8	0	false
de novo biosynthesis of pyrimidine ribonucleotides	19	18	0	false
fatty acid oxidation pathway	11	10	0	false
gluconeogenesis	22	22	2	false
glutamate degradation I	6	6	2	true
glutamate fermentation I-the hydroxyglutarate pathway	9	8	0	false
glycerol degradation II	16	16	2	false
glycolysis I	17	17	2	false
heme biosynthesis II	15	14	0	false
histidine biosynthesis I	19	18	0	false
homocysteine and cysteine interconversion	7	8	2	true
homoserine and methionine biosynthesis	14	13	2	false
homoserine biosynthesis	5	4	0	false
isoleucine biosynthesis I	9	8	0	false
isoleucine degradation III	17	16	0	false
leucine biosynthesis	7	6	0	false
lipoxygenase pathway	13	12	6	false
mannosyl-chito-dolichol biosynthesis	5	4	0	false
methionine biosynthesis I	8	7	2	false
methionine biosynthesis III	5	4	0	false
non-oxidative branch of the pentose phosphate pathway	10	12	4	true
polyamine biosynthesis I	8	7	0	false
polyamine biosynthesis III	8	7	0	false
pyridine nucleotide biosynthesis	10	9	2	false
pyridine nucleotide cycling	15	18	6	true
pyruvate oxidation pathway	6	5	2	false
riboflavin and FMN and FAD biosynthesis	17	17	2	true
salvage pathways of purine and pyrimidine nucleotides	29	35	13	true
salvage pathways of purine nucleosides	16	17	4	true
salvage pathways of pyrimidine ribonucleotides 1	5	4	0	false
salvage pathways of pyrimidine ribonucleotides 2	7	6	2	false
serine biosynthesis	5	4	0	false
serine-isocitrate lyase pathway	28	29	4	true
spermine biosynthesis	7	6	0	false

sucrose biosynthesis	8	7	2	false
sucrose degradation I	5	4	0	false
sucrose degradation III	12	12	4	false
superpathway of fatty acid oxidation and glyoxylate cycle 1	7	6	0	false
superpathway of fatty acid oxidation and glyoxylate cycle 2	11	10	0	false
superpathway of glycolysis and TCA variant VIII	46	55	16	true
superpathway of glycolysis, pyruvate dehydrogenase, TCA, and glyoxylate bypass	39	41	4	true
superpathway of isoleucine and valine biosynthesis 1	7	6	0	false
superpathway of isoleucine and valine biosynthesis 2	9	8	0	false
superpathway of leucine, valine, and isoleucine biosynthesis 1	14	13	2	false
superpathway of lysine, threonine and methionine biosynthesis	9	8	0	false
superpathway of phenylalanine, tyrosine and tryptophan biosynthesis	25	24	0	false
superpathway of ribose and deoxyribose phosphate degradation 1	9	8	2	false
superpathway of ribose and deoxyribose phosphate degradation 2	8	7	0	false
superpathway of serine and glycine biosynthesis	7	6	0	false
superpathway of sulfur amino acid biosynthesis	19	20	4	true
TCA cycle – aerobic respiration	19	20	2	true
TCA cycle variation VIII	28	32	8	true
threonine biosynthesis	9	8	0	false
trehalose biosynthesis III	7	6	0	false
tryptophan biosynthesis	11	10	0	false
UDP-N-acetylgalactosamine biosynthesis	9	8	0	false
urate degradation	12	11	2	false
ureide degradation	9	9	6	false
valine biosynthesis	7	6	0	false
xylulose-monophosphate cycle	10	10	0	true

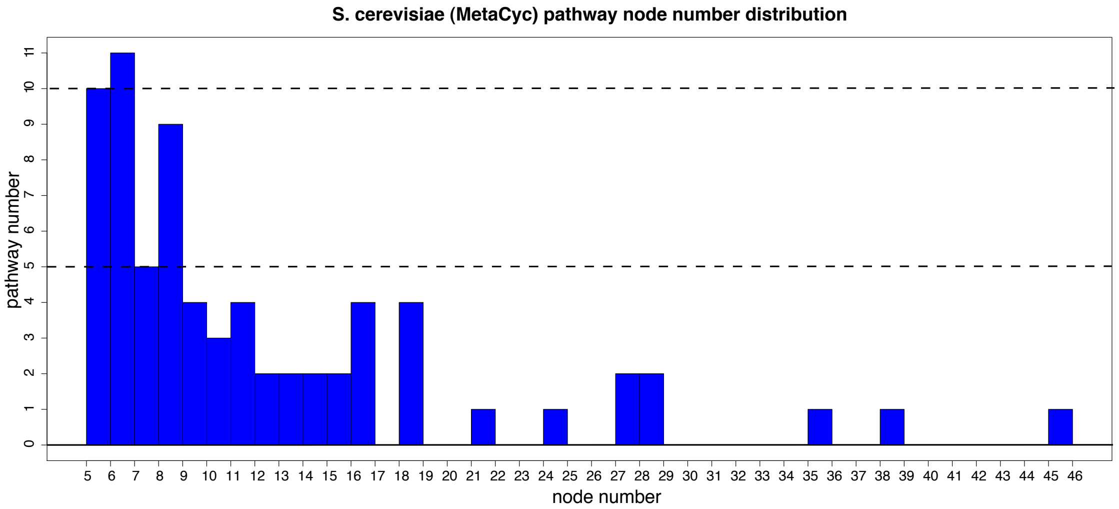


Figure 9.5: Histogram of pathway node numbers for the 71 reference MetaCyc pathways used for the evaluation of multiple-end pathway prediction. All pathways below five nodes have been discarded.

9.2.3 Comparison of a predicted to a reference pathway

In order to quantify the accuracy of the metabolic pathway prediction algorithms, predicted pathways have to be compared with annotated ones.

Accuracy calculation

In this thesis, accuracies are calculated as in [31, 32] based on the overlap of nodes in the predicted and annotated pathway.

A *true positive* (TP) node is a node that is present in both the reference and the predicted pathway. A *false positive* (FP) node is a node that is present in the predicted but absent in the reference pathway. Finally, a *false negative* (FN) node is a node that is absent in the predicted but present in the reference pathway.

Importantly, seed nodes are not counted as true positives.

Figure 9.6 illustrates these definitions.

The *sensitivity* of pathway prediction is then defined as:

$$Sn = \frac{TP}{TP+FN} \tag{9.1}$$

The *positive predictive value* is defined as:

$$PPV = \frac{TP}{TP+FP} \tag{9.2}$$

The PPV instead of the specificity is used for accuracy calculation, because the formula for *specificity* involves true negatives:

$$Spec = \frac{TN}{TN+FP} \tag{9.3}$$

However, the true negatives in the case of pathway prediction are all compounds and reactions present in the database that were absent from both predicted and annotated pathway. This is a huge number (∼18,000 for the KEGG reaction network) with respect to the number of false positives, thus $Spec$ would be always close to one.

Given Sn and PPV, the *geometric accuracy* is defined as:

$$Acc_g = Sn * PPV \tag{9.4}$$

The *arithmetic accuracy* $A_a = \frac{Sn+PPV}{2}$ is avoided, because in extreme cases, it is dominated by either Sn or PPV. Consider for instance a subgraph that includes the entire KEGG LIGAND reaction graph. It would have a sensitivity of one and a PPV close to zero. Still, its overall arithmetic accuracy would be as high as 0.5, whereas its geometric accuracy would be close to zero.

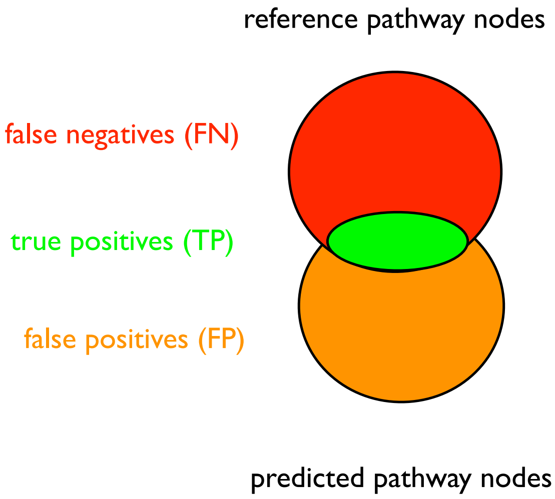


Figure 9.6: The accuracy of pathway prediction is calculated based on the overlap of the nodes in the reference and predicted pathway. False negatives are nodes that are present in the reference pathway but absent from the predicted pathway (red). False positives are non-seed nodes that are present in the predicted but absent from the reference pathway (orange), whereas true positives are non-seed nodes that are present in both the reference and the predicted pathway (green intersection of reference and predicted node set).

Node set overlap versus pathway alignment

An alternative to the node set comparison would be a pathway alignment. However, pathway alignment algorithms would have comparatively long runtimes, as they have to cope with branched pathways. Thus, their use is infeasible in a large-scale evaluation such as the one presented in chapter 3. The main advantage of pathway alignment algorithms is their consideration of node arrangement. However, because of the structure of metabolic pathways, the nodes composing them cannot be arranged in any number of ways. Thus, node set intersections reflect well the degree of similarity between two metabolic pathways.

Overall, the accuracy measurement is quite conservative. Since terminal compounds were removed from the reference pathways, they are counted as false positives. Likewise, alternative reactions between two compounds are counted as false positives, even though they share a reactant pair with the correct reaction.

9.2.4 Evaluation procedure

Before starting the evaluation, the reference pathway set, the metabolic graph, the pathway prediction algorithm to be evaluated and its parameters are selected. Then, the evaluation iterates over the set of reference pathways. The final accuracy is the average of all individual accuracies.

The results of the evaluations are stored in a database. For two-end pathway prediction, a web interface has been written, so that the results can be browsed on the internet at: http://rsat.ulb.ac.be/pathfindingsupplementref/index.html.

Evaluation of path finding

For each reference pathway, the following steps are performed:

1. The terminal reactions of the reference pathway are selected as seed nodes.

2. The metabolic graph and the seeds are given to the pathway prediction algorithm, which returns a pathway.

3. The accuracy of the predicted pathway is computed as described above.

Evaluation of multiple-end pathway prediction

For each reference pathway, the following steps are performed:

1. All terminal reactions of the reference pathway are selected as seed nodes. Pathway prediction and accuracy computation are carried out for these seeds.

2. One additional intermediate reaction is selected at random from the reference pathway. Pathway prediction and accuracy computation are repeated with the terminal and intermediate reactions as seed nodes.

3. Step 2 is repeated as many times as reactions exist in the reference pathway. Reactions already in the seed node set are not selected again from the reference pathway.

4. The evaluation of the reference pathway terminates if all its reactions have been added to the seed node set.

Figure 9.7 shows the result of the evaluation for the Takahashi-Matsuyama algorithm in the directed, degree-weighted MetaCyc graph.

9.3 Metabolic database

In order to store, manage and access metabolic data from different sources, a database is needed.

In particular, the database has to meet the three following requirements:

- Storage and retrieval of metabolic networks obtained from different source databases.
- Storage and retrieval of metabolic pathways obtained from different source databases.
- Mapping of genes to reactions and reactant pairs.

The aMAZE database [101] would have been the database of choice, but it is no longer maintained. Other generic biological or metabolic databases or parsers [99, 25, 58] were not available at the time or did not offer needed functionality (such as coverage of KEGG RPAIR).

Therefore, it became necessary to design and implement a metabolic database that performs the tasks listed above.

9.3.1 Data model

The data model of the metabolic database was inspired by the aMAZE data model [163, 101], but is much simpler, since its goals are much less ambitious than those of the aMAZE database. The aMAZE database was designed to accommodate biological data in general (metabolism, regulation, signal transduction, etc.), whereas the database presented here is specific to metabolic data and does not cover regulation or signal transduction. Furthermore, it does not model explicitly the stoichiometry of reactions, because this was not needed for the pathway prediction approach adopted in this thesis. Polypeptides that are themselves reactants of reactions cannot be modeled, because the database is restricted to small molecule metabolism. Care has been taken to ensure that organism-specific features of pathways such as reaction directions are modeled at the level of pathway steps and not at the level of reactions, compounds and enzymes. The pathwayStep class roughly corresponds to the catalysis class in biopax, with the difference that pathwayStep objects can be associated to main compounds. Proteins consisting of several polypeptides or operons comprising several genes can be modeled with the help of the bioentity child and parent classes.

Figure 9.8 depicts the data model of the metabolic database.

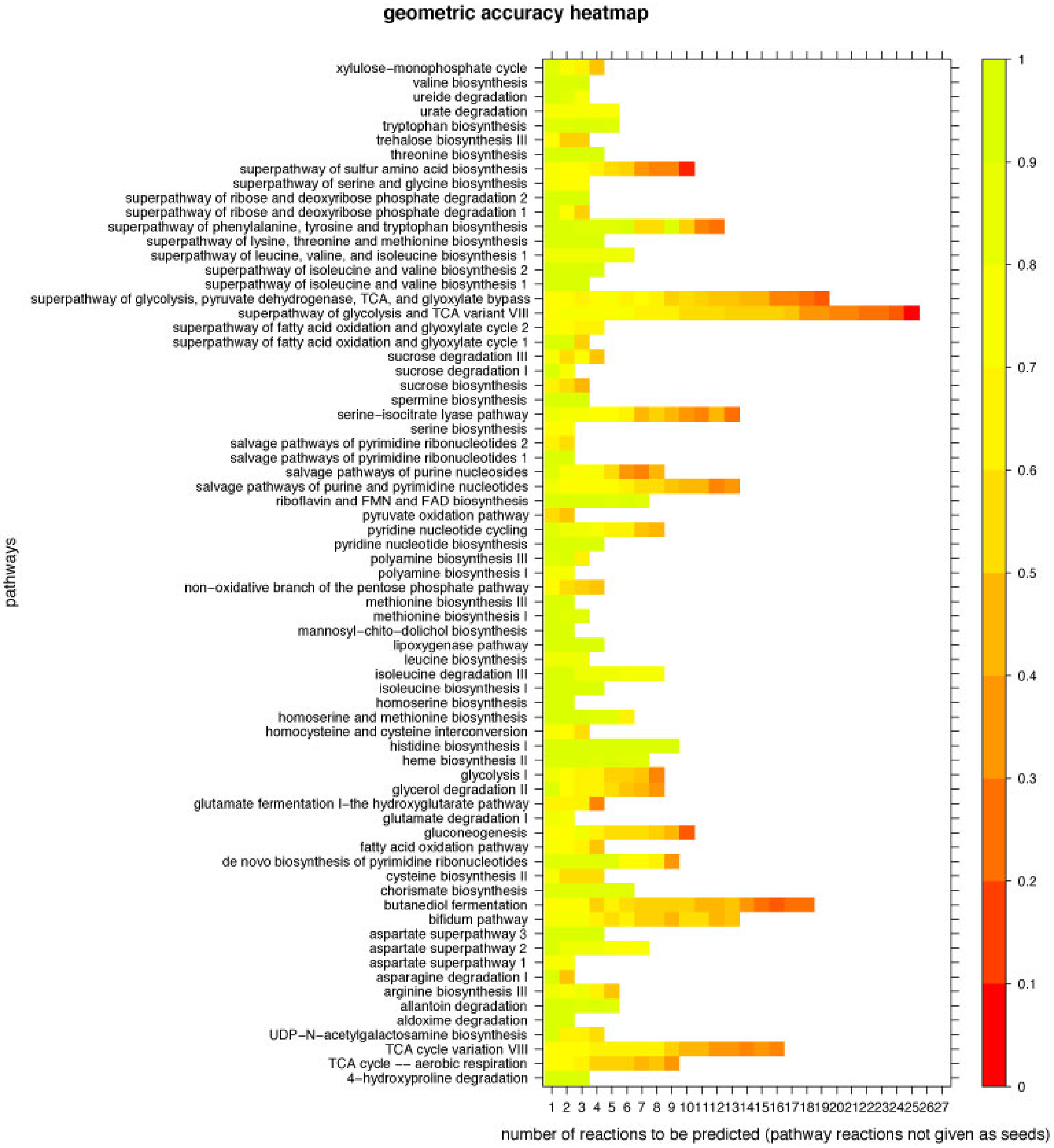


Figure 9.7: This heat map summarizes the evaluation performed for the algorithm Takahashi-Matsuyama in the directed, degree-weighted MetaCyc graph. The x-axis lists the number of reaction nodes that still have to be predicted. Thus, for x equals 1, all except one reaction were given as seed nodes to the algorithm. For x equals 2, all but two reactions were given as seeds to the algorithm, and so forth. The y-axis enumerates all 71 reference pathways in alphabetical order. The colors of the heat map reflect the geometric accuracy, with red for an accuracy of 0, orange for an accuracy of 0.5 and green for an accuracy of 1. Overall, 406 predictions were carried out.

Table 9.5: Content of the metabolic database as of October 2009.

Database	version	Number of reactions/ reactant pairs	Number of compounds	Number of pathways	Number of organisms
KEGG LIGAND	49.0	7,432	6,078	0	0
KEGG RPAIR	49.0	10,913	5,661	0	0
KEGG PATHWAY	50.0	5,301	4,233	145	1 (reference)
MetaCyc	13.0	4,116	3,412	1347	391
aMAZE	2006 (no longer maintained)	392	443	116	3

9.3.2 Implementation

The metabolic database is object oriented. The underlying relational database is hosted by postgres [135] and the object-relational mapping was implemented in Java with Hibernate [71]. These technologies have the advantage to be open source (postgres: BSD license, Hibernate: GNU Lesser General Public License) and can therefore be freely distributed.

9.3.3 Contents of the metabolic database

Since most metabolic databases provide their data in biopax format [16], a parser was written to load data in this format into the metabolic database. In addition, a parser for KGML files was written, because at the time these files were not available in biopax format (they have been converted recently to biopax). These parsers do not modify or filter the data, except for removing orphan compounds, non-small molecule compounds (such as polymers and glycans) and reactions having identical substrates and products or involving non-small molecules. To implement more sophisticated quality checks was out of the scope of this thesis.

Currently, the metabolic database contains KEGG LIGAND, KEGG RPAIR, the reference maps of KEGG PATHWAY, MetaCyc and aMAZE pathways.

Since the parsers are included in the NeAT command line tool set, users can update the data or add more data, e.g. organism-specific KEGG PATHWAY maps or PGDBs (pathway/genome databases) from BioCyc (such as HumanCyc).

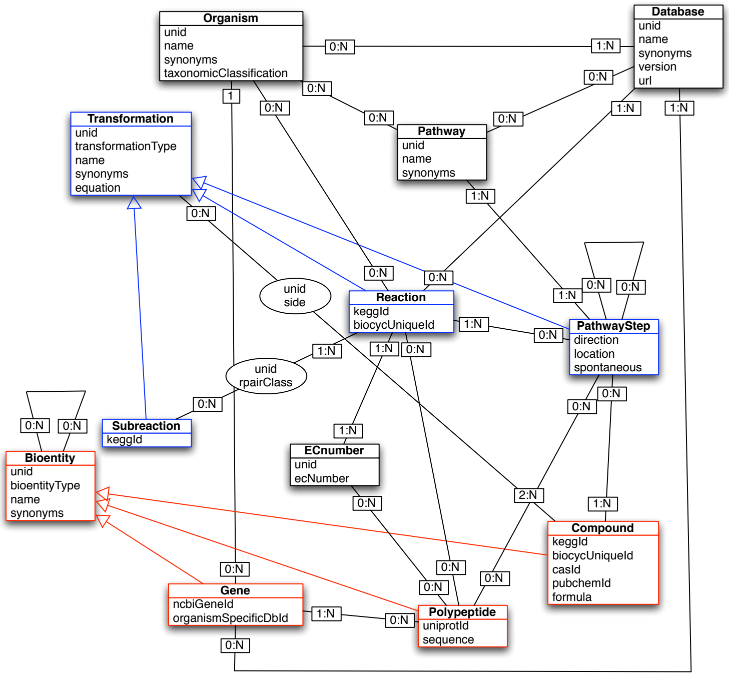


Figure 9.8: Data model of the metabolic database, showing the classes with their attributes and the relations between them.

A Introduction to graph theory

In this section, a number of concepts and definitions from graph theory that are important for this thesis are explained. Alternative names for the defined terms are given in brackets. Most of the definitions have been adapted from [67]. Figure A.2 visualizes some of these concepts.

Graph

A graph is a structure consisting of a set of nodes (also called vertices) and a set of edges. An edge is defined as an unordered pair of nodes (the nodes of a pair do not need to be distinct in case of a self-loop). The unordered node pair represents the end points of the edge.

Directed graph (digraph)

The definition of a directed graph is the same as of a graph, except that the node pair that represents the end points of an edge is now ordered. The first node of the ordered pair is also called *tail node*, the second *head node*. Thus, an edge points from the tail node to the head node. An edge in a directed graph is also called an *arc*. From this definition follows that an arc has always one of two possible directions. For instance, given the nodes u and v, an arc may point from u to v or from v to u.

Simple graph

A simple graph contains no self-loops or multiple edges (arcs) between a node pair. All graphs used in this thesis are simple.

Bipartite graph

A bipartite graph (digraph) consists of two node sets A and B and one edge (arc) set. Each edge (arc) has one end point belonging to A and the other end point belonging to B. Thus, there is no edge (arc) between any two nodes of the same node set.

Hypergraph

A hypergraph is a graph where each edge is an unordered node set. A graph is a special case of a hypergraph where each edge is a node set with only two (possibly identical) nodes. A directed hypergraph is a directed graph where each arc is an ordered pair of (possibly empty) disjoint node sets A and B, where A is the tail of the arc and B the head. Thus, the hypergraph is a generalized graph where an edge (arc) may connect more than two nodes. Figure A.1 shows an example of a hypergraph.

Weighted graph

A weighted graph (digraph) is a (directed) graph where each edge (arc) is assigned a real number, called its weight or cost. A variant is the node-weighted graph (digraph), where each node instead of each edge (arc) receives a weight.

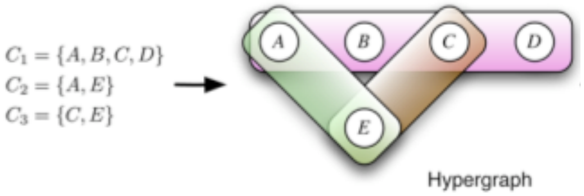


Figure A.1: The three node sets C1, C2 and C3 define three hyperedges in a hypergraph. The Figure was taken from Figure 1 in [93].

Table A.1: Adjacency matrix of graph shown in Figure A.2.

	1a	1b	1c	1d	1e	2a	2b	2c	2d
1a	0	0	0	0	0	3	2	0	0
1b	0	0	0	0	0	0	0	0	0
1c	0	0	0	0	0	0	0	3	0
1d	0	0	0	0	0	0	0	0	4
1e	0	0	0	0	0	0	0	0	0
2a	0	2	0	6	0	0	0	0	0
2b	0	0	0	66	0	0	0	0	0
2c	0	0	0	0	0	0	0	0	0
2d	0	0	0	0	0	0	0	0	0

Subgraph

A subgraph of a graph (digraph) G is a graph H whose nodes and edges (arcs) are in G. The *weight of a subgraph* is the sum of its edge (arc) weights. In a node-weighted graph, it is the sum of its node weights.

Adjacency matrix

The adjacency matrix uniquely describes a graph or digraph. Given a graph G with n nodes, its adjacency matrix A is a $n \times n$ square matrix, where each entry A_{ij} may be either 0 (there is no edge between node i and j) or 1 (there is an edge between node i and j). In a weighted graph, entry A_{ij} may be a real number, which represents the weight of the edge between node i and j. The adjacency matrix of a graph is always symmetric, but it may be asymmetric in case of a digraph.

As an example, the adjacency matrix of the directed, weighted, bipartite graph depicted in Figure A.2 is given in Table A.1.

In-degree

In a directed graph D, the in-degree of a node v is the number of arcs in D of which it is a head node.

Out-degree

In a directed graph D, the out-degree of a node v is the number of arcs in D of which it is a tail node.

Degree (connectivity)

The degree of a node v in a graph is the number of edges of which it is an end point plus twice the number of self loops. In a directed graph, the degree of a node v is the sum of its in- and out-degree.

Neighbor nodes (neighbors)

A neighbor of a node A is an end point of an edge, of which node A is the other end point.

Walk

In a graph, a walk from node v_0 to node v_n is an alternating sequence $W =< v_0, e_1, v_1, e_2, ..., v_{n-1}, e_n, v_n >$ of nodes and edges such that the endpoints of edge e_i is the node pair v_{i-1},v_i for $i = 1, ..., n$. In a directed graph, W is a walk if each arc e_i is directed from node v_{i-1} to node v_i, that is v_{i-1} is the tail of arc e_i and v_i is the head of arc e_i for all $i = 1, ..., n$. Importantly, a node v or an edge (arc) e may appear more than once in a walk. A *trivial walk* consists of only one node and no edges (arcs). A *closed walk* is a non-trivial walk that begins and ends with the same node. An *open walk* begins and ends with different nodes.

Path (Simple path)

A path is a walk with no repeated edges (arcs) and no repeated nodes (except the start and end node in a closed path). The *weight of a path* is the sum of its edge (arc) weights. In a node-weighted graph, it is the sum of its node weights. The *length of a path* is the number of its edges (arcs).

Cycle

A non-trivial closed path is called a cycle.

Terminal nodes

In this thesis, terminal nodes are nodes in a directed graph that have either an in-degree of zero or an out-degree of zero.

Node distance

In [31, 32], the distance between two nodes u and v in a node-weighted graph is defined as: $dist(u, v) = w(shortest\_path(u, v)) - \frac{w(u)+w(v)}{2}$, where w denotes the path weight or node weight.

Connected graph

A node v is reachable from a node u if there is a walk from u to v. A graph is connected if for every pair of nodes u and v, there is a walk from u to v.

Weakly connected digraph (Connected digraph)

A digraph is weakly connected if for every pair of nodes u and v, there is a walk from u to v and/or a walk from v to u.

Strongly connected digraph

Two nodes u and v are mutually reachable in a digraph D, if D contains both a directed u-v walk and a directed v-u walk. A digraph is strongly connected if every two of its nodes are mutually reachable. A strongly connected digraph is also weakly connected.

Components

A graph G that is not connected is made up of components, each of which is a connected subgraph of G.

Orphan node (orphan)

In this thesis, a trivial walk is called orphan node. An orphan node has a degree of zero.

Tree

A tree is a connected graph (or weakly connected digraph) that has no cycles.

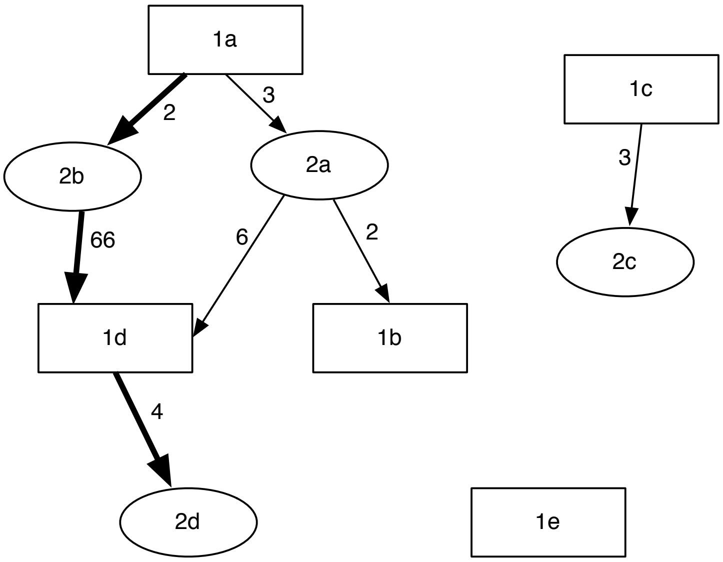


Figure A.2: A directed, weighted, bipartite graph D is depicted. It has two node sets, namely set *square* $= \{1a, 1b, 1c, 1d, 1e\}$ and set *ellipse* $= \{2a, 2b, 2c, 2d\}$. Each arc has one end point belonging to *square* and the other end point belonging to *ellipse*. Each arc is in addition labeled with an integer, its weight. Graph D consists of three weakly connected components H1, H2 and H3, where H1 contains the nodes 1a, 2a, 2b, 1b, 1d and 2d, H2 the nodes 1c and 2c and H3 the node 1e. Node 1e is an orphan node. Node 2a has an in-degree of one, an out-degree of two and therefore a degree of three. Thick arcs depict a possible path from start node 1a to end node 2d. The weight of this path is the sum of its arc weights, namely 72. Subgraph H1, H2 and H3 form each a tree, but graph D is not a tree, since it is not connected.

Bibliography

[1] P. Adler, J. Reimand, J. Jänes, R. Kolde, H. Peterson, and J. Vilo. KEGGanim: pathway animations for high-throughput data. *Bioinformatics*, 24:588–590, 2008. 1

[2] E. Almaas, Z.N. Oltvai, and A-L. Barabási. The activity reaction core and plasticity of metabolic networks. *PLoS Computational Biology*, 1:e68, 2005. 21

[3] A. Antonov, S. Dietmann, and H.W. Mewes. KEGG spider: interpretation of genomics data in the context of the global gene metabolic network. *Genome Biology*, 9, 2008. 50, 52, 54, 169

[4] A. Antonov, S. Dietmann, P. Wong, and H.W. Mewes. TICL - a web tool for network-based interpretation of compound lists inferred by high-throughput metabolomics. *FEBS Journal*, 276:2084–2094, 2009. 50, 54

[5] M. Arita. Metabolic reconstruction using shortest paths. *Simulation Practice and Theory*, 8:109–125, 2000. 43

[6] M. Arita. In Silico Atomic Tracing by Substrate-Product Relationships in Escherichia coli Intermediary Metabolism. *Genome Research*, 13:2455–2466, 2003. 36, 37, 39, 43, 45, 164

[7] M. Arita. The metabolic world of Escherichia coli is not small. *PNAS*, 101:1543–1547, 2004. 26, 32, 36, 39, 42, 43

[8] G.D. Bader, D. Betel, and C.W.V. Hogue. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Research*, 31:248–250, 2003. 51

[9] A. Bairoch. The ENZYME data bank. *Nucleic Acids Research*, 22:3626–3627, 1994. 25

[10] V. Batagelj and A. Mrvar. *Graph Drawing Software*, chapter Pajek - Analysis and Visualization of Large Networks, pages 77–103. Mathematics and Visualization. Springer, Berlin, 2004. 134

[11] J.E. Beasley and F.J. Planes. Recovering metabolic pathways via optimization. *Bioinformatics*, 23:92–98, 2007. 33, 36, 37, 48

[12] A. Beloqui, M.-E. Guazzaroni, F. Pazos, J.M. Vieites, M. Godoy, O.V. Golyshina, T.N. Chernikova, A. Waliczek, R. Silva-Rocha, Y. Al-ramahi, V. La Cono, C. Mendez, J.A.

Salas, R. Solano, M.M. Yakimov, K.N. Timmis, P.N. Golyshin, and M. Ferrer. Reactome array: Forging a link between metabolome and genome. *Science*, 326:252–257, 2009. 171

[13] J.M. Berg, J.L. Tymoczko, and L. Stryer. *Biochemistry, fifth edition*. W.H. Freeman and Company, 2002. 3

[14] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000. 10

[15] N. Betzler. Steiner Tree Problems in the Analysis of Biological Networks. Master's thesis, Universität Tübingen, 2006. 53, 176

[16] BioPAX: Biological Pathways Exchange. http://www.biopax.org/. 134, 193

[17] T. Blum and O. Kohlbacher. MetaRoute: fast search for relevant metabolic routes for interactive network navigation and visualization. *Bioinformatics*, 24:2108–2109, 2008. 46

[18] T. Blum and O. Kohlbacher. Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks. *Journal of Computational Biology*, 15:565–576, 2008. 32, 36, 37, 39, 45, 162, 164

[19] F. Boyer and A. Viari. Ab initio reconstruction of metabolic pathways. *Bioinformatics*, 19:ii26–ii34, 2003. 36, 43

[20] S. Brohée, K. Faust, G. Lima-Mendez, O. Sand, R. Janky, G. Vanderstocken, Y. Deville, and J. van Helden. NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Research*, 36:W444–W451, 2008. 132

[21] J. Callut. *First Passage Times Dynamics in Markov Models with Applications to HMM Induction, Sequence Classification, and Graph Mining*. PhD thesis, Université catholique de Louvain, 2007. 54, 134, 177

[22] R. Caspi, H. Foerster, C.A. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S.Y. Rhee, A.G. Shearer, C. Tissier, T.C. Walk, P. Zhang, and P.D. Karp. The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Research*, 36:D623–D631, 2008. 4, 11, 24, 31

[23] F. Centler, P. Speroni di Fenizio, N. Matsumaru, and P. Dittrich. Chemical Organizations in the central sugar metabolism of Escherichia coli. Modeling and Simulation in Science Engineering and Technology, Post-proceedings of ECMTB 2005, 2005. 28

[24] F. Centler and P. Dittrich. Chemical organizations in atmospheric photochemistries: a new method to analyze chemical reaction networks. *Planet Space Science*, 2006. 28

[25] E.G. Cerami, G.D. Bader, B.E. Gross, and C. Sander. cPath: open source software for collecting, storing, and querying biological pathways. *BMC Bioinformatics*, 7:497, 2006. 191

[26] A. Chang, M. Scheer, A. Grote, I. Schomburg, and D. Schomburg. Brenda, amenda and frenda the enzyme information system: new content and tools in 2009. *Nucleic Acids Research*, 37:D588–D592, 2009. 25

[27] C.-H. Chou, W.-C. Chang, C.-M. Chiu, C.-C. Huang, and H.-D. Huang. FMM: a web server for metabolic pathway reconstruction and comparative analysis. *Nucleic Acids Research*, 37:W129–W134, 2009. 36, 39, 42

[28] N. Christian, P. May, S. Kempa, T. Handorf, and O. Ebenhöh. An integrative approach towards completing genome-scale metabolic networks. *Mol. Biosyst.*, Advance Access, 2009. 31

[29] F. Couche. Recherche de chemins sur les voies métaboliques. Technical report, Université Libre de Bruxelles, 2002. 42

[30] D. Croes. *Recherches de chemins dans le réseau métabolique et mesure de la distance métabolique entre enzymes*. PhD thesis, Université Libre de Bruxelles, 2005. 29, 35

[31] D. Croes, F. Couche, S. Wodak, and J. van Helden. Metabolic pathfinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Research*, 33:W326–W330, 2005. 17, 32, 33, 36, 39, 42, 43, 45, 57, 133, 188, 197

[32] D. Croes, F. Couche, S. Wodak, and J. van Helden. Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.*, 356:222–236, 2006. 17, 32, 33, 36, 37, 39, 42, 45, 57, 133, 188, 197

[33] M. Csete and J. Doyle. Bow ties, metabolism and disease. *TRENDS in Biotechnology*, 22, 2004. 21

[34] 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*, 31:19–20, 2002. 1

[35] F. Darvas. MetabolExpert, an expert system for predicting updated version of the compact procedure for the evaluation metabolism of substances. *QSAR in Environmental Toxicology*, 81:709–737, 1987. 49

[36] T. Davidsen, E. Beck, A. Ganapathy, R. Montgomery, N. Zafar, Q. Yang, R. Madupu, P. Goetz, K. Galinsky, O. White, and G. Sutton. The comprehensive microbial resource. *Nucleic Acids Research*, page Advance Access, 2009. 4

[37] L.F. de Figueiredo, A. Podhorski, A. Rubio, C. Kaleta, J.E. Beasley, S. Schuster, and F.J. Planes. Computing the shortest elementary flux modes in genome-scale metabolic networks. *Bioinformatics*, page Advanced access, 2009. 47, 125

[38] L.F. de Figueiredo, S. Schuster, C. Kaleta, and D.A. Fell. Can sugars be produced from fatty acids? A test case for pathway analysis tools. *Bioinformatics*, 24:2615–2621, 2008. 47, 121, 123

[39] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 175

[40] M.T. Dittrich, G.W. Klau, A. Rosenwald, T. Dandekar, and T. Müller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, 24:i223–i231, 2008. 50, 51, 53, 170

[41] D.J.Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998. 20

[42] G. Dooms, Y. Deville, and P. Dupont. Constrained path finding in biochemical networks. In *In 5émes Journées Ouvertes Biologie Informatique Mathématiques*, 2004. 38

[43] G. Dooms, Y. Deville, and P. Dupont. Constrained metabolic network analysis: Discovering pathways using cp(graph), 2005. 38

[44] The DOT Language. http://www.graphviz.org/doc/info/lang.html. 133, 135, 169

[45] S.E. Dreyfus and R.A. Wagner. The steiner tree problem in graphs. *Networks*, 1:195–207, 1971. 50, 53

[46] N.C. Duarte, S.A. Becker, N. Jamshidi, I. Thiele, M.L. Mo, T.D. Vo, and R. Srivas ad B.Ø. Palsson. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *PNAS*, 104:1777–1782, 2007. 31, 169

[47] C.W. Duin, A. Volgenant, and S. Voß. Solving group steiner problems as steiner problems. *European Journal of Operational Research*, 154:323–329, 2004. 133, 173, 175

[48] P. Dupont, J. Callut, G. Dooms, J.-N. Monette, and Y. Deville. Relevant subgraph extraction from random walks in a graph. *Research Report (Scientifique - portée internationale)*, 2006-2007. 54, 83, 134, 177

[49] O. Ebenhöh, T. Handorf, and R. Heinrich. Structural analysis of expanding metabolic networks. *Genome Informatics*, 15:35–45, 2004. 27

[50] L. Ellis, J. Gao, K. Fenner, and L.P. Wackett. The University of Minnesota pathway prediction system: predicting metabolic logic. *Nucleic Acids Research*, 36:W427–W432, 2008. 11, 36, 37, 49, 170

[51] D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28:652–673, 1999. 40, 172

[52] K. Faust, S. Brohée, G. Lima-Mendez, G. Vanderstocken, and J. van Helden. Network Analysis Tools: from biological networks to clusters and pathways. *Nature Protocols*, 3:1616–1629, 2009. 132

[53] K. Faust, D. Croes, and J. van Helden. In response to "Can sugars be produced from fatty acids? A test case for pathway analysis tools". *Bioinformatics*, Advanced Access, 2009. 47

[54] K. Faust, D. Croes, and J. van Helden. Metabolic pathfinding using RPAIR annotation. *J. Mol. Biol.*, 388:390–414, 2009. 32, 36, 37, 39, 43, 45, 46

[55] K. Faust, P. Dupont, J. Callut, and J. van Helden. Pathway discovery in metabolic networks by subgraph extraction. Submitted, 2009. 50, 52, 54

[56] D.A. Fell and A. Wagner. The small world of metabolism. *Nature Metabolic Engineering*, 18:1121–1122, 2000. 36, 42

[57] K. Fenner, J. Gao, S. Kramer, L. Ellis, and L. Wackett. Data-driven extraction of relative reasoning rules to limit combinatorial explosion in biodegradation pathway prediction. *Bioinformatics*, 24:2079–2085, 2008. 36, 49

[58] F. Le Fèvre, S. Smidtas, and V. Schächter. Cyclone: java-based querying and computing with pathway/genome databases. *Bioinformatics*, 23:1299–1300, 2007. 191

[59] C.V. Forst and K. Schulten. Evolution of metabolisms: A new method for the comparison of metabolic pathways using genomics information. *Journal of Computational Biology*, 6:343–360, 1999. 23

[60] J. Gagneur, D.B. Jackson, and G. Casari. Hierarchical analysis of dependency in metabolic networks. *Bioinformatics*, 19:1027–1034, 2003. 165

[61] S. Gama-Castro, V. Jiménez-Jacinto, M. Peralta-Gil, A. Santos-Zavaleta, M.I. Pe naloza Spinola, B. Contreras-Moreira, J. Segura-Salazar, L. Mu niz Rascado, I. Martínez-Flores, H. Salgado, C. Bonavides-Martínez, C. Abreu-Goodger, C. Rodríguez-Penagos, J. Miranda-Ríos, E. Morett, E. Merino, A.M. Huerta, L. Trevi no Quintanilla, and J. Collado-Vides. RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic acids research*, 36, 2008. 4, 11, 165

[62] P. Gerlee, L. Lizana, and K. Sneppen. Pathway identification by network pruning in the metabolic network of E. coli. *Bioinformatics*, page Advance Access, 2009. 165

[63] P. Godard, A. Urrestarazu, S. Vissers, K. Kontos, G. Bontempi, J. van Helden, and B. Andre. Effect of 21 Different Nitrogen Sources on Global Gene Expression in the Yeast Saccharomyces cerevisiae. *Molecular and Cellular Biology*, 27:3065–3086, 2007. 92, 93, 94, 116, 160

[64] N. Goffard and G. Weiller. PathExpress: a web-based tool to identify relevant pathways in gene expression data. *Nucleic Acids Research*, 35:W176–W181, 2007. 1

[65] M.L Green and P.D. Karp. A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. *BMC Bioinformatics*, 5:76, 2004. 31

[66] N. Greene, P.N. Judson, J.J. Langowski, and C.A. Marchant. Knowledge-based Expert Systems for Toxicity and Metabolism Prediction: DEREK, StAR, and METEOR. *SAR and QSAR in Environmental Research*, 10:299–313, 1999. 49

[67] J.L. Gross and J. Yellen. *Graph theory and its applications, second edition*. Discrete mathematics and its applications. Chapman and Hall, CRC, 2 edition, 2006. 175, 195

[68] R. Guimerà and L.A.N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005. 20, 165

[69] V. Hatzimanikatis, C. Li, J.A. Ionita, C.S. Henry, M.D. Jankowski, and L.J. Broadbelt. Exploring the diversity of complex metabolic networks. *Bioinformatics*, 21:1603–1609, 2005. 49

[70] J.J. Heijnen. Approximative kinetic formats used in metabolic network modeling. *Biotechnology and Bioengineering*, 91:534–545, 2005. 164

[71] HIBERNATE. Relational Persistence for Java and .NET. https://www.hibernate.org/. 133, 193

[72] M. Himsolt. GML: A portable Graph File Format. http://www.infosun.fim.uni-passau.de/Graphlet/GML/gml-tr.html. 133, 135

[73] C. Hold and S. Panke. Towards the engineering of in vitro systems. *Journal of the Royal Society Interface*, 6:S507–S521, 2009. 126

[74] E.L. Hong, R. Balakrishnan, Q. Dong, K.R. Christie, J. Park, G. Binkley, M.C. Costanzo, S.S. Dwight, S.R. Engel, D.G. Fisk, J.E. Hirschman, B.C. Hitz, C.J. Krieger, M.S. Livstone, S.R. Miyasato, R.S. Nash, R. Oughtred, M.S. Skrzypek, S. Weng, E.D. Wong, K.K. Zhu, K. Dolinski, D. Botstein, and J.M. Cherry. Gene Ontology annotations at SGD: new data sources and annotation methods. *Nucleic Acids Research*, 36:D577–D581, 2008. 10, 106

[75] B.K. Hou, L.B.M. Ellis, and L.P. Wackett. Encoding microbial metabolic logic: predicting biodegradation. *J. Ind. Microbiol. Biotechnol.*, 31:261–272, 2004. 37

[76] Z. Hu, D.M. Ng, T. Yamada, C. Chen, S. Kawashima, J. Mellor, B. Linghu, M. Kanehisa, J.M. Stuart, and C. DeLisi. VisANT 3.0: new modules for pathway visualization, editing, prediction and construction. *Nucleic Acids Research*, 35:W625–W32, 2007. 53, 133, 135

[77] T. Ideker, O. Ozier, B. Schwikowski, and A.F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18:S233–S240, 2002. 49, 50, 51, 53, 169, 170

[78] T. Ideker, V. Thorsson, J.A. Ranish, R. Christmas, J. Buhler, J.K. Eng, R. Bumgarner, D.R. Goodlett, R. Aebersold, and L. Hood. Integrated Genomic and Proteomic Analyses of a Systematically Perturbed Metabolic Network. *Science*, 292:929–934, 2001. 51

[79] J. Ihmels, R. Levy, and N. Barkai. Principles of transcriptional control in the metabolic network of Saccharomyces cerevisiae. *Nature biotechnology*, 22:86–92, 2004. 167

[80] J. Jaworska, S. Dimitrov, N. Nikolova, and O. Mekenyan. Probabilistic assessment of biodegradatability based on metabolic pathways: CATABOL system. *SAR and QSAR in Environmental Research*, 13:307–323, 2002. 36, 49, 170

[81] L.J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic, P. Bork, and C. von Mering. STRING 8 - a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Research*, 37:D412–D416, 2009. 135

[82] H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai, and A.-L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407:651–654, 2000. 17, 19, 35, 42

[83] V.M. Jimenez and A. Marzal. Computing the k shortest paths: a new algorithm and an experimental comparison. *Lecture Notes in Computer Science - Proceedings of the 3rd International Workshop on Algorithm Engineering*, 1668:15–29, 1999. 40, 57, 133, 172

[84] B. H. Junker, C. Klukas, and F. Schreiber. VANTED: A system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics*, 7, 2006. 1

[85] C. Kaleta, L.F. de Figueiredo, and S. Schuster. Can the whole be less than the sum of its parts? Pathway analysis in genome-scale metabolic networks using elementary flux patterns. *Genome Research*, 19:1872–1883, 2009. 47, 125

[86] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, 36:D480–D484, 2008. 2, 4, 5, 6, 8, 9, 11

[87] P.D. Karp, S. Paley, and P. Romero. The pathway tools software. *Bioinformatics*, 18:S225–S232, 2002. 13, 31

[88] P.D. Karp and S.M. Paley. Representations of metabolic knowledge: Pathways. In *Proc Int Conf Intell Syst Mol Biol*, volume 2, pages 203–211, 1994. 5, 26

[89] R.M. Karp. *Reducibility among combinatorial problems*, pages 85–103. Complexity of Computer Computations. R. E. Miller and J. W. Thatcher, Plenum Press, 1972. 175

[90] I.M. Keseler, C. Bonavides-Martínez, J. Collado-Vides, S. Gama-Castro, R.P. Gunsalus, D.A. Johnson, M. Krummenacker, L.M. Nolan, S. Paley, I.T. Paulsen, M. Peralta-Gil, A. Santos-Zavaleta, A.G. Shearer, and P.D. Karp. EcoCyc: a comprehensive view of Escherichia coli biology. *Nucleic Acids Research*, 37:D464–D470, 2009. 3, 42

[91] R. Khanin and E. Wit. How scale-free are biological networks. *Journal of Computational Biology*, 13:810–818, 2006. 17, 20

[92] P. Kharchenko, D. Vitkup, and G.M. Church. Filling gaps in a metabolic network using expression information. *Bioinformatics*, 20:i178–i185, 2004. 31, 37

[93] S. Klamt, U.-U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS Computational Biology*, 5:5, 2009. 16, 196

[94] S. Klamt, J. Saez-Rodriguez, and E.D. Gilles. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Systems Biology*, 1:2, 2007. 37

[95] P. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees, 1993. 53, 175

[96] M. Kotera, M. Hattori, M.-A. Oh, R. Yamamoto, T. Komeno, J. Yabuzaki, K. Tonomura, S. Goto, and M. Kanehisa. RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions. *Genome Informatics*, 15:P062, 2004. 3, 13, 43, 45

[97] M. Kotera, Y. Okuno, M. Hattori, S. Goto, and M. Kanehisa. Computational Assignment of the EC Numbers for Genomic-Scale Analysis of Enzymatic Reactions. *Journal of the American Chemical Society*, 126:16487–16498, 2004. 43, 55

[98] R. Küffner, R. Zimmer, and T. Lengauer. Pathway analysis in metabolic databases via differential metabolic display. *Bioinformatics*, 16:825–836, 2000. 16, 25

[99] J. Küntzer, C. Backes, T. Blum, A. Gerasch, M. Kaufmann, O. Kohlbacher, and H.-P. Lenhof. BNDB - The Biochemical Network Database. *BMC Bioinformatics*, 8:367, 2007. 191

[100] I. Lee, S.V. Date, A.T. Adai, and E.M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, 2004. 167

[101] C. Lemer, E. Antezana, F. Couche, F. Fays, X. Santolaria, R. Janky, Y. Deville, J. Richelle, and S. Wodak. The aMAZE LightBench: a web interface to a relational database of cellular processes. *Nucleic Acids Research*, 32:D444–D448, 2004. 40, 43, 55, 191

[102] W. Li and H. Kurata. A grid layout algorithm for automatic drawing of biochemical networks. *Bioinformatics*, 21:2036–2042, 2005. 169

[103] V.A. Likić. Databases of metabolic pathways. *BIOCHEMISTRY AND MOLECULAR BIOLOGY EDUCATION*, 34:408–412, 2006. 13

[104] G. Lima-Mendez and J. van Helden. The powerful law of the power law and other myths in network biology. *Molecular BioSystems*, Advance access, 2009. 20

[105] I. Ljubić, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Math. Program. Ser. B*, 105:427–449, 2006. 50, 53, 182

[106] Y. Lu, J. A, G. Wang, H. Hao, Q. Huang, B. Yan, W. Zha, S. Gu, H. Ren, Y. Zhang, X. Fan, M. Zhang, and K. Hao. Gas chromatography/time-of-flight mass spectrometry based metabonomic approach to differentiating hypertension- and age-related metabolic variation in spontaneously hypertensive rats. *Rapid Commun Mass Spectrom*, 22:2882–2888, 2008. 170

[107] H. Ma and A.-P. Zeng. Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics*, 19:270–277, 2003. 20, 42

[108] W.K. Maas. The Arginine Repressor of Escherichia coli. *MICROBIOLOGICAL REVIEWS*, 58:631–640, 1994. 11

[109] N. Matsumaru, F. Centler, P. Speroni di Fenizio, and P. Dittrich. Chemical organization theory applied to virus dynamics. *Information Technology*, 48:154–160, 2006. 28

[110] V. Matys, O.V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A.E. Kel, and E. Wingender. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34:D108–D110, 2006. 51

[111] M. L. Mavrovouniotis. Identification of qualitatively feasible metabolic pathways. *Artificial intelligence and molecular biology*, pages 325–364, 1993. 31, 32, 33, 36, 46, 168

[112] D.C. McShan, S. Rao, and I. Shah. PathMiner: predicting metabolic pathways by heuristic search. *Bioinformatics*, 19(13):1692–1698, 2003. 36, 38, 43

[113] A. Mithani, G. M. Preston, and J. Hein. Hypergraph based tool for metabolic pathway prediction and network comparison. *Bioinformatics*, 25:1831–1832, 2009. 16, 32, 36, 39, 45

[114] Y. Moriya, M. Itoh, S. Okuda, A.C. Yoshizawa, and M. Kanehisa. KAAS: an automatic genome annotation and pathway reconstruction server. *Nucleic Acids Research*, 35:W182–W185, 2007. 13, 31

[115] J. Moult, K. Fidelis, A. Kryshtafovych, B. Rost, T. Hubbard, and A. Tramontano. Critical assessment of methods of protein structure prediction-round vii. *Proteins*, 69:3–9, 2007. 170

[116] K. Moutselos, I. Kanaris, A. Chatziioannou, I. Maglogiannis, and F.N. Kolisis. KEGGconverter: a tool for the in-silico modelling of metabolic networks of the KEGG pathways database. *BMC Bioinformatics*, 10:324, 2009. 133

[117] D.L. Nelson and M.M. Cox. *Lehninger Principles Of Biochemistry, fourth edition*. Worth Publishers, 2005. 23

[118] I.E. Nikerel, W.A. van Winden, P.J.T. Verheijen, and J.J. Heijnen. Model reduction and a priori kinetic parameter identifiability analysis using metabolome time series for metabolic reaction networks with linlog kinetics. *Metabolic Engineering*, 11:20–30, 2009. 126, 164

[119] J. Noirel, S. Y. Ow, G. Sanguinetti, A. Jaramillo, and P. C. Wright. Automated extraction of meaningful pathways from quantitative proteomics data. *Briefings in Functional Genomics and Proteomics*, 7:136–146, 2008. 49, 50, 51, 53

[120] J.F. Nyc, H.K. Mitchell, E. Leifer, and W.H. Langham. The use of isotopic carbon in a study of the metabolsim of anthranilic acid in neurospora. *The Journal of Biological Chemistry*, 179:783–787, 1949. 30

[121] International Union of Biochemistry and Molecular Biology Nomenclature Committee. Enzyme Nomenclature 1992: IUB Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes, 1992. 7

[122] M.A. Ott and G. Vriend. Correcting ligands, metabolites, and pathways. *BMC Bioinformatics*, 7:517, 2006. 166

[123] S. M. Paley and P. D. Karp. The Pathway Tools cellular overview diagram and Omics Viewer. *Nucleic Acids Research*, 34:3771–3778, 2006. 1

[124] S.M. Paley and P.D. Karp. Evaluation of Computational Metabolic-Pathway Predictions for Helicobacter pylori. *Bioinformatics*, 18:715–724, 2002. 31

[125] J.A. Papin, J. Stelling, N.D. Price, S. Klamt, S. Schuster, and B.Ø. Palsson. Comparison of network-based pathway analysis methods. *TRENDS in Biotechnology*, 22:400–405, 2004. 28, 47, 48

[126] B. Papp, C. Pal, and L.D. Hurst. Metabolic network analysis of the causes and evolution of enzyme dispensability in yeast. *PLoS Computational Biology*, 429:661–664, 2004. 22

[127] F. Pazos, D. Guijas, A. Valencia, and V. De Lorenzo. Metarouter: bioinformatics for bioremediation. *Nucleic Acids Research*, 35:D588–D592, 2005. 37, 38

[128] J.M. Peregrín-Alvarez, C. Sanford, and J. Parkinson. The conservation and evolutionary modularity of metabolism. *Genome Biology*, 10:R63, 2009. 20, 21

[129] T. Pfeiffer, I. Sanchez-Valdenebro, J.C. Nuno, F. Montero, and S. Schuster. METATOOL: for studying metabolic networks. *Bioinformatics*, 15:251–257, 1999. 28, 32, 33, 36, 47

[130] E. Pitkänen, P. Jouhten, and J. Rouso. Inferring branching pathways in genome-scale metabolic networks. *BMC Systems Biology*, 3:103, 2009. 164

[131] E. Pitkänen, A. Rantanen, J. Rousu, and E. Ukkonen. Finding feasible pathways in metabolic networks. In *In Proceedings of the 10th Panhellenic Conference on Informatics (PCI'2005), Lecture Notes in Computer Science*, pages 123–133. Springer, 2005. 16, 27, 28

[132] F.J. Planes and J. Beasley. A critical examination of stoichiometric and path-finding approaches to metabolic pathways. *Briefings in Bioinformatics*, 9:422–436, 2008. 23, 47, 48

[133] F.J. Planes and J.E. Beasley. An optimisation model for metabolic pathways. *Bioinformatics*, 25:2723–2729, 2009. 33, 36, 37, 48, 126

[134] M.G. Poolman, B.K. Bonde, A. Gevorgyan, H.H. Patel, and D.A. Fell. Challenges to be faced in the reconstruction of metabolic networks from public databases. *IEE Proc.-Syst. Biol.*, 153:379–384, 2006. 166

[135] PostgreSQL. The world's most advanced open source database. http://www.postgresql.org/. 193

[136] T.S.K. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, L. Balakrishnan, A. Marimuthu, S. Banerjee, D.S. Somanathan, A. Sebastian, S. Rani, S. Ray, C.J.H. Kishore, S. Kanth, M. Ahmed, M.K. Kashyap, R. Mohmood, Y.L. Ramachandra, V. Krishna, B.A. Rahiman, S. Mohan, P. Ranganathan, S. Ramabadran, R. Chaerkady, and A. Pandey. Human Protein Reference DatabaseÑ2009 update. *Nucleic Acids Research*, 37:D767–D772, 2009. 51

[137] J. Quackenbush. Microarrays - Guilt by Association. *Science*, 302:240–241, 2003. 1

[138] S.A. Rahman, P. Advani, R. Schunk, R. Schrader, and D. Schomburg. Metabolic pathway analysis web service (pathway hunter tool at cubic). *Bioinformatics*, 2004. 32, 36, 39, 43, 162

[139] D. Rajagopalan and P. Agarwal. Inferring pathways from gene lists using a literature-derived network of biological relationships. *Bioinformatics*, 21(5):788–793, 2005. 49, 50, 51, 53, 170

[140] E. Ravasz, A.L. Somera, D.A. Mongru, and Z.N. Oltvai, A.-L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551–1555, 2002. 20, 21

[141] M.J. Herrgård, N. Swainston, P. Dobson, W.B. Dunn, K.Y. Arga, M. Arvas, N. Blüthgen, S. Borger, R. Costenoble, M. Heinemann, M. Hucka, N. Le Novère, P. Li, W. Liebermeister, M.L. Mo, A.P. Oliveira, D. Petranovic, S. Pettifer, E. Simeonidis, K. Smallbone, I. Spasié, D. Weichart, R. Brent, D.S. Broomhead, H.V. Westerhoff, B. Kürdar, M. Penttilä, E. Klipp, B.Ø. Palsson, U. Sauer, S.G. Oliver, P. Mendes, J. Nielsen, and D.B. Kell. A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nature Biotechnology*, 26:1155–1160, 2008. 120

[142] A. Rosenwald, G. Wright, W.C. Chan, J.M. Connors, E. Campo, and et al. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *N. Engl. J. Med.*, 346:1937–1947, 2002. 51

[143] S. Schuster, T. Dandekar, and D.A. Fell. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *TIBTECH*, 17:53–60, 1999. 28

[144] S. Schuster, T. Pfeiffer, F. Moldenhauer, I. Koch, and T. Dandekar. Exploring the pathway structure of metabolism: decomposition into subnetworks and application. *Bioinformatics*, 18:351–361, 2002. 20

[145] R. Schwarz, C. Liang, C. Kaleta, M. Kühnel, E. Hoffmann, S. Kuznetsov, M. Hecker, G. Griffiths, S. Schuster, and T. Dandekar. Integrated network reconstruction, visualization and analysis using YANAsquare. *BMC Bioinformatics*, 8:313, 2007. 133

[146] R. Schwarz, P. Musch, A. von Kamp, B. Engels, H. Schirmer, S. Schuster, and T. Dandekar. YANA - a software tool for analyzing flux modes, gene-expression and enzyme activities. *BMC Bioinformatics*, 6:135, 2005. 37

[147] M.S Scott, T. Perkins, S. Bunnell, F. Pepin, D.Y. Thomas, and M. Hallett. Identifying regulatory subnetworks for a set of genes. *Mol Cell Proteomics*, 4(5):683–692, 2005. 50, 51, 53, 176

[148] A. Seressiotis and J.E. Bailey. Mps : An algorithm and data base for metabolic pathway synthesis. *Biotechnology Letters*, 8:837–842, 1986. 36, 46

[149] A.S.N Seshasayee, G.M. Fraser, M.M. Babu, and N.M. Luscombe. Principles of transcriptional regulation and evolution of the metabolic system in E. coli. *Genome Research*, 19:79–91, 2008. 167

[150] 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*, 13:2498–2504, 2003. 32

[151] T. Shlomi, M.N. Cabili, M.J. Herrgård, B.Ø. Palsson, and E. Ruppin. Network-based prediction of human tissue-speciÞc metabolism. *Nature Biotechnology*, 26:1003–1010, 2008. 166

[152] 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(2):S219–S230, 2002. 36, 42

[153] K. Stensjö, S.Y. Ow, M.E. Barrios-Llerena, P. Lindblad, and P.C. Wright. An iTRAQ-based quantitative analysis to elaborate the proteomic response of Nostoc sp. PCC 7120 under N2 fixing conditions. *J. Proteome Research*, 6:621–635, 2001. 51, 170

[154] H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Math. Japonica*, 24:573–577, 1980. 54, 84, 134, 175, 176

[155] E.L. Tatum and D. Bonner. Indole and serine in the biosynthesis and breakdown of tryptophane. *PNAS*, 30:30–37, 1944. 30

[156] E.L. Tatum, D. Bonner, and G.W. Beadle. Anthranilic acid and the biosynthesis of indole and tryptophan by Neurospora. *Arch. Biochem.*, 3:477, 1944. 30

[157] M. Terzer and J. Stelling. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24:2229–2235, 2008. 47, 125

[158] B. Testa, A.-L. Balmat, and A. Long. Predicting drug metabolism: Concepts and challenges. *Pure Appl. Chem.*, 76:907–914, 2004. 37

[159] O. Thimm, O. Bläsing, Y. Gibon, A. Nagel, S. Meyer, P. Krüger, J. Selbig, L. A. Müller, 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*, 37:914–939, 2004. 1

[160] R. Tibshirani. A simple method for assessing sample sizes in microarray experiments. *BMC Bioinformatics*, 7:106, 2006. 92

[161] C.T. Trinh, P. Unrean, and F. Srienc. Minimal Escherichia coli Cell for the Most Efficient Production of Ethanol from Hexoses and Pentoses. *Applied and Environmental Microbiology*, 74:3634–3643, 2008. 33, 37

[162] C.T. Trinh, A. Wlaschin, and F. Srienc. Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism. *Appl. Microbiol. Biotechnol.*, 81:813–826, 2009. 37, 47

[163] 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*, 381:921–935, 2000. 191

[164] J. van Helden, L. Wernisch, D. Gilbert, and S. Wodak. Graph-based analysis of metabolic networks. In *Ernst Schering Research Foundation Workshop.*, volume 38, pages 245–274. Springer-Verlag, 2002. 16, 36, 42

[165] I. Vastrik, P. D'Eustachio, E. Schmidt, G. Joshi-Tope, G. Gopinath, D. Croft, B. de Bono, M. Gillespie, B. Jassal, S. Lewis, L. Matthews, G. Wu, E. Birney, and L. Stein. Reactome: a knowledge base of biologic pathways and processes. *Genome Biology*, 8:R39, 2007. 11, 169

[166] A. von Kamp and S. Schuster. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics*, 22:1930–1931, 2006. 37, 47

[167] C. von Mering, E.M. Zdobnov, S. Tsoka, F.D. Ciccarelli, J.B. Pereira-Leal, C.A. Ouzounis, and P. Bork. Genome evolution reveals biochemical networks and functional modules. *PNAS*, 100:15428–15433, 2003. 167

[168] D. Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28:31–36, 1988. 5, 45

[169] U. Wittig and A. De Beuckelaer. Analysis and comparison of metabolic pathway databases. *Briefings in Bioinformatics*, 2:126–142, 2001. 13

[170] K.H. Wong, M.J. Hynes, and M.A. Davis. Recent advances in nitrogen regulation: a comparison between yeast and filamentous fungi. *Eukaryotic cell*, 7:917–925, 2008. 107

[171] Y. Yamanishi. Supervised enzyme network inference from the integration of genomic data and chemical information. *Bioinformatics*, 21:i468–i477, 2005. 49

[172] Y. Yamazaki, M. Kitajima, M. Arita, H. Takayama, H. Sudo, M. Yamazaki, N. Aimi, and K. Saito. Biosynthesis of Camptothecin. In Silico and in Vivo Tracer Study from [1-13C]Glucose. *PLANT PHYSIOLOGY*, 134:161–170, 2004. 30, 37

[173] Ch. Yang, D.A. Rodionov, X. Li, O.N. Laikova, M.S. Gelfand, O.P. Zagnitko, M.F. Romine, A.Y. Obraztsova, K.H. Nealson, and A.L. Osterman. Comparative Genomics and Experimental Characterization of N-Acetylglucosamine Utilization Pathway of Shewanella oneidensis. *The Journal of Biological Chemistry*, 281:29872–29885, 2006. 30

[174] J.D. Zhang and S. Wiemann. KEGGgraph: a graph approach to KEGG PATHWAY in R and Bioconductor. *Bioinformatics*, 25:1470–1471, 2009. 133

[175] J. Zhao, H. Yu, J.-H. Luo, Z.-W. Cao, and Y.-X. Li. Hierarchical modularity of nested bow-ties in metabolic networks. *BMC Bioinformatics*, 7:386, 2006. 21

[176] J. Zhu and M.Q. Zhang. SCPD: a promoter database of the yeast Saccharomyces cerevisiae. *Bioinformatics*, 15:607–611, 1999. 51

[177] A. Zien, R. Küffner, R. Zimmer, and T. Lengauer. Analysis of gene expression data with pathway scores. In *Proceedings of the International Conference of Intelligent Systems Molecular Biology*, pages 407–417, 2000. 49