Analysis and Visualization of Metabolic Networks: A Hypergraph Approach

by

Evaggelia M. Maniadi

Thesis submitted in partial fulfilment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete

School of Sciences and Engineering

Computer Science Department

October 2011

University of Crete
Computer Science Department

**Analysis and Visualization of Metabolic Networks:
A Hypergraph Approach**

Thesis submitted by
Evaggelia M. Maniadi
in partial fulfilment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Evaggelia M. Maniadi

Committee approvals: _____
Ioannis Tollis
Professor, Thesis Supervisor

_____
Konstantinos Marias
Principal Researcher, FORTH-ICS, Committee Member

_____
Ioannis Tzitzikas
Assistant Professor, Committee Member

Departmental approval: _____
Angelos Bilas
Associate Professor, Chairman of Graduate Studies

Heraklion, October 2011

Abstract

Every biological organism consists of cells. Cells are complex, biological systems whose growth and existence depends on thousands of biological interactions between the molecules (genes, proteins, chemical compounds etc). These interactions are typically modelled as graphs, where nodes represent the molecules and edges the interactions between the molecules. Many important biological networks have been discovered through laboratory studies. To date, at least four types of biological networks have been characterized in detail: the gene regulatory networks, the protein-protein interaction networks, the signal transduction networks and the metabolic interaction networks.

This thesis focuses on the analysis and visualization of metabolic networks. Metabolic networks represent the set of chemical reactions that happen in a cell to maintain life. Biochemical reactions are processes that lead to the transformation of one set of chemical substances (*called reactants or substrates*) to another set (*called products*). A variety of problems arise in analysing and representing that knowledge using a graph-based model. For example, by treating each metabolite in a reaction separately, dependences between metabolites are lost. Since a reaction may have one or more substrates or one or more products, the edge representing a reaction is actually a hyperedge. Thus, the metabolism is better represented by hypergraphs. A hypergraph is a generalization of an ordinary graph where a hyperedge can connect more than two vertices.

Unfortunately, the hypergraph based analysis and visualization of metabolic networks was proven not a simple task, due to the complexity and the particularities of metabolic networks and due to the lack of literature on hypergraph drawing. For testing our approach a tool, called VisBolic (*VISualization of MetaBOLIC pathways*), has been developed. The tool analyses and visualizes metabolic pathways while allows the execution of predefined but meaningful queries

on metabolism. The metabolic network data come from KEGG database, where the data are stored in flat files. Therefore, the data were firstly preprocessed and then were stored in a relational database, making them amenable to computer analysis.

Περίληψη

Κάθε οργανισμός αποτελείται από κύτταρα. Τα κύτταρα είναι πολύπλοκα, βιολογικά συστήματα των οποίων η ύπαρξη και η ανάπτυξη εξαρτάται από χιλιάδες αλληλεπιδράσεις μεταξύ των μορίων του κυττάρου (πχ γονίδια, πρωτεΐνες, χημικές ενώσεις κτλ). Αυτές οι αλληλεπιδράσεις συνήθως μοντελοποιούνται με τη βοήθεια γράφων, όπου οι κόμβοι αναπαριστούν τα μόρια ενώ οι ακμές τις αλληλεπιδράσεις μεταξύ των μορίων. Μέχρι σήμερα έχουν μελετηθεί λεπτομερώς τουλάχιστον τέσσερα διαφορετικά είδη βιολογικών δικτύων: gene-regulatory networks, protein-protein interaction networks, signal transduction networks και metabolic interaction networks.

Η παρούσα εργασία επικεντρώνεται στην ανάλυση και οπτικοποίηση των metabolic networks. Τα metabolic networks αναπαριστούν το σύνολο των αντιδράσεων που λαμβάνουν χώρα σε ένα κύτταρο με σκοπό την διατήρηση της ζωής. Οι βιο-χημικές αντιδράσεις είναι διαδικασίες που οδηγούν στη μετατροπή ενός συνόλου από χημικά συστατικά σε ένα άλλο. Το αρχικό σύνολο περιλαμβάνει τα *αντιδρώντα* και το δεύτερο τα *προϊόντα* της αντίδρασης. Η αναπαράσταση των metabolic networks με την χρήση γράφω συχνά επιφέρει διάφορα προβλήματα. Για παράδειγμα, σε μία αντίδραση αντιμετωπίζοντας κάθε μεταβολίτη ξεχωριστά, οι εξαρτήσεις μεταξύ των μεταβολιτών χάνονται. Καθώς μία αντίδραση μπορεί να έχει ένα ή περισσότερα αντιδρώντα ή και ένα ή περισσότερα προϊόντα, η ακμή που αναπαριστά την αντίδραση είναι στην πραγματικότητα μία υπερακμή και το metabolic network ένας υπεργράφος. Ένας υπεργράφος είναι μια γενίκευση ενός απλού γράφου, όπου μία υπερακμή μπορεί να συνδέει έναν ή περισσότερους κόμβους.

Δυστυχώς η ανάλυση και η οπτικοποίηση των metabolic networks με την χρήση υπεργράφων δεν είναι απλή διαδικασία. Η δυσκολία οφείλεται στην πολυπλοκότητα και στην ιδιαιτερότητα των δικτύων αυτών καθώς επίσης και στην περιορισμένη βιβλιογραφία αναφορικά με την οπτικοποίηση υπεργράφων. Για την εξέταση του προτεινόμενου μοντέλου, αναπτύχθηκε ένα εργαλείο, το VisBolic (*VISualization of MetaBOLIC pathways*), το οποίο αναλύει και οπτικοποιεί τα metabolic

networks ενώ επιτρέπει την εκτέλεση προκαθορισμένων επερωτήσεων στη βάση. Τα δεδομένα προέρχονται από τη βάση δεδομένων του KEGG, και βρίσκονται αποθηκευμένα σε flat αρχεία. Τα δεδομένα αυτά, αφού υπέστησαν μία αρχική επεξεργασία, στη συνέχεια αποθηκεύτηκαν σε μία σχεσιακή βάση δεδομένων, καθιστώντας εφικτή την ανάλυσή τους με υπολογιστή.

Ευχαριστίες

Καταρχήν, θα ήθελα να ευχαριστήσω τον επόπτη καθηγητή μου κ. Ιωάννη Τόλλη για την πολύτιμη καθοδήγηση και συμβολή του για την ολοκλήρωση αυτής της εργασίας. Επίσης τον ευχαριστώ για την εμπιστοσύνη που μου έδειξε κατά τη διάρκεια της συνεργασίας μας.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τον καθηγητή κ. Γιάννη Τζίτζικα και τον ερευνητή κ. Κώστα Μαριά για την προθυμία τους να συμμετάσχουν στην τριμελή επιτροπή για την εξέταση της μεταπτυχιακής μου εργασίας.

Παράλληλα θα ήθελα να ευχαριστήσω το Ίδρυμα Τεχνολογίας και Έρευνας και συγκεκριμένα το εργαστήριο Υπολογιστικής Ιατρικής του Ινστιτούτου Πληροφορικής για την οικονομική ενίσχυση και την υλικοτεχνική υποδομή που μου παρείχε όλο αυτό το διάστημα.

Τέλος ένα μεγάλο ευχαριστώ στους φίλους μου για την υποστήριξή τους καθώς και στην οικογένειά μου για τη στήριξη, την ενθάρυνση και την αγάπη τους όλο αυτό το διάστημα.

Contents

1 Introduction **1**

1.1 Metabolic Networks 3

1.2 Basics in Graphs 7

1.3 Motivation 9

2 Existing Databases and Models **11**

2.1 Existing databases 11

2.1.1 The KEGG database 13

2.1.2 EC Numbers 15

2.1.3 Summary 15

2.2 Existing Graph-based models 17

2.2.1 Compound or Substrate Graphs 18

2.2.2 Reaction Graphs 19

2.2.3 Enzyme Graphs 19

2.2.4 Bipartite and Hypergraphs 20

3 Proposed Model and Database **23**

3.1 Why not compound graphs? 23

3.2 Why not bipartite graphs? 25

3.3 Proposed Model 27
3.4 Hypergraph Drawings 28
3.5 VisBolic Database 31
3.6 Treatment of cofactors in analysis 35

4 Analysis of Metabolic Pathways and Networks 39

4.1 Parameters related to Shortest Paths 40
4.2 Centrality Measures 40
4.2.1 Degree Centrality 40
4.2.2 Betweenness Centrality 41
4.2.3 Closeness Centrality 42
4.3 Parameters related to Neighbourhood 42
4.3.1 Neighbourhood Connectivity 42
4.3.2 Network Density 43
4.3.3 Shared Neighbours 44
4.3.4 Topological Coefficients 44
4.4 Clustering Coefficient 44
4.5 Graph Analysis Algorithms 45

5 Visualization of Metabolic Pathways 49

5.1 Related Work 51
5.2 Proposed Method 53
5.2.1 Treatment of cofactors in visualization 55
5.3 Results 55

6 Conclusions and Future Directions 81

Bibliography **85**

Appendices **95**

A VisBolic Manual **95**

A.1 Editor - Edit Mode 96

A.2 Editor - Navigational Mode 99

A.3 The menu-bar 99

A.3.1 File menu 99

A.3.2 View menu 100

A.3.3 Select menu 101

A.3.4 Query menu 102

A.3.5 Analysis menu 102

A.3.6 Layout menu 110

A.3.7 Help 112

A.4 The tool-bar 112

A.5 Future Work 112

B A comparison of visualization tools **115**

List of Figures

1.1 Simplified UML diagram of the various objects involved in metabolism 5

1.2 Alternative pathways for methionine biosynthesis in E.coli and S. cerevisiae . . . 8

2.1 The TCA cycle or Krebs cycle or Citrate cycle of the bacterium Escherichia coli K-12 DH10B as presented in KEGG database. The circles represent metabolites whereas the green boxes depict the enzymes that catalyse the corresponding reactions. 14

2.2 Classification of Glycolysis in KEGG, EcoCyc and WIT database. The figure is borrowed from [80]. 16

2.3 The connectivity of the main pathway database types. The figure is borrowed from [80]. 17

2.4 Reaction: $glutamate + ATP \longrightarrow gamma - glutamylphosphate + ADP$. The figure is borrowed from [24]. 18

2.5 Ambiguity in existing models. The figure is borrowed from [24]. 19

2.6 Reaction and Enzyme graph for the sets of reactions: R1: A + B $\rightarrow$ C + D, R2: D + E $\rightarrow$ F + G, R3: F + G $\rightarrow$ H + I, and R4: I $\rightarrow$ J + K. E1, E2 and E3 are enzymes: E1 catalyses R1 and R4, E2 catalyses R2 and E3 catalyses R3. (a) Reaction graph (b) Enzyme graph 20

2.7 Metabolic reaction as given in many biochemical textbooks. This kind of edge is called hyperedge. 21

2.8 Graphical representation of an undirected hypergraph with V = { v1, v2, v3, v4, v5, v6, v7 }, E = { e1, e2, e3, e4 } = { { v1, v2, v3 } , { v2, v3 } , { v3, v5, v7 } , { v4 } }. Source: `http://en.wikipedia.org/wiki/Hypergraph`. 21

3.1 Example of undirected hypergraph arising in the context of biological network analysis. The figure is borrowed from [52]. 25

3.2 Example of directed hypergraph arising in the context of biological network analysis. The figure is borrowed from [52]. 26

3.3 (a) Example of reactions and (b) their directed graph representation (introducing the notion of hypernodes). The figure is taken from [60]. 28

3.4 Example of a simple, hypothetical metabolic pathway. (a) Its adjacency list representation; (b) its adjacency matrix representation. Underlined metabolites determine main compounds. 29

3.5 Four drawings of the hypergraph H = (V,E) with V = $\{1, 2, 3, 4, 5, 6\}$ and E = $\{\{1, 2\}, \{1, 2, 3\}, \{2, 4\}, \{3, 4, 5, 6\}\}$: (a) subset-based drawing; (b) edge-based drawing; (c) Zykov representation; (d) bipartite graph. The figure is borrowed from [50]. 30

3.6 A hyperedge based Venn diagram for (a) K_5; (b) $K_{3,3}$; (c) two different drawings of a vertex based Venn diagram for H with V = $\{1, 2, 3, 4, 5, 6\}$ and E = $\{\{1, 2, 3, 4\}, \{1, 2, 3, 5\}, \{1, 2, 3, 6\}\}$. The figure is borrowed from [50]. 31

3.7 Example entry from KEGG 32

3.8 EER Model of the VisBolic database containing its main relations (diagram created with mySQL Workbench) 34

4.1 (a) Hypergraph with hypercycles (marked in bold) (b) Directed graph with cycles (marked in bold) 47

5.1 KEGG example 50

5.2 ExPASy 50

5.3 Visualization of Propanoate metabolism using Becker's and Rojas's algorithm . 52

5.4 2 1/2D drawing of seven related pathways (parallel projection) 52

5.5 An example of a reaction - Zero size dummy nodes are inserted at the forking positions. 53

5.6 Cases of inserting one dummy vertex: (a) The reaction has one reactant and multiple products; (b) The reaction has multiple reactants and one product. The small white circles represent the dummy vertices. 54

5.7 (a) Case of inserting two dummy vertices. The reaction has multiple reactants and multiple products. The small white circles represent the dummy vertices; (b) Case of inserting no dummy vertex. 54

5.8 Drawing of Thiamine metabolism of Bacillus subtilis with hierarchical layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 17 reactions and 29 compounds. The drawing contains one edge crossing. 56

5.9 Drawing of Thiamine metabolism of Bacillus subtilis with hierarchical layout algorithm using **CellDesigner**. The circles represent reactions whereas the green boxes depict the metabolites and enzymes that catalyse the corresponding reactions. 57

5.10 Drawing of Glycolysis / Gluconeogenesis of Bacillus subtilis with circular layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 30 reactions and 46 compounds. The drawing is planar. 58

5.11 Drawing of Glycolysis / Gluconeogenesis of Bacillus subtilis with circular layout algorithm using **VANTED**. The circles represent metabolites whereas the green boxes depict the enzymes that catalyse the corresponding reactions. 59

5.12 Drawing of Lysine biosynthesis of Escherichia coli K-12 DH10B with organic (smart) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 12 reactions and 29 compounds. The drawing is planar. 60

5.13 Drawing of Lysine biosynthesis of Escherichia coli K-12 DH10B with organic (smart) layout algorithm using **CellDesigner**. The circles represent reactions whereas the green boxes depict the metabolites and enzymes that catalyse the corresponding reactions. 61

5.14 Drawing of Pyruvate metabolism of Saccharomyces cerevisiae (budding yeast) with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 24 reactions and 45 compounds. The drawing is planar. 62

5.15 Drawing of Pyruvate metabolism of Saccharomyces cerevisiae (budding yeast) with organic (classic) layout algorithm using **Cytoscape**. The circles represent reactions whereas the rhombuses depict the metabolites and enzymes that catalyse the corresponding reactions. 63

5.16 Drawing of Pyruvate metabolism of Saccharomyces cerevisiae (budding yeast) with organic (classic) layout algorithm using **CellDesigner**. The circles represent reactions whereas the green boxes depict the metabolites and enzymes that catalyse the corresponding reactions. 64

5.17 Drawing of Nicotinate and nicotinamide metabolism of Helicobacter pylori P12 with orthogonal (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 10 reactions and 23 compounds. 65

5.18 Drawing of Pyrimidine metabolism of Caulobacter crescentus CB15 with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 38 reactions and 49 compounds. 67

5.19 Drawing of One carbon pool by folate of Caulobacter crescentus CB15 with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 17 reactions and 33 compounds. 68

5.20 A part of the drawing of Purine metabolism of Caenorhabditis elegans (nematode) with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 70 reactions and 78 compounds. 69

5.21 Drawing of Valine, leucine and isoleucine degradation of Caenorhabditis elegans (nematode) with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 34 reactions and 55 compounds. 70

5.22 Drawing of Alanine, aspartate and glutamate metabolism of Chlamydomonas reinhardtii with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 17 reactions and 40 compounds. 71

5.23 Drawing of Nitrogen metabolism of Chlamydomonas reinhardtii with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 10 reactions and 29 compounds. 72

5.24 Drawing of Fatty acid elongation in mitochondria of Drosophila melanogaster (fruit fly) with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 25 reactions and 34 compounds. 73

5.25 Drawing of One carbon pool by folate of Drosophila melanogaster (fruit fly) with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 17 reactions and 32 compounds. 74

5.26 Drawing of Purine metabolism of Synechocystis sp. PCC6803 with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 66 reactions and 77 compounds. 75

5.27 Drawing of Pyrimidine metabolism of Tetrahymena thermophila with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 42 reactions and 43 compounds. 76

5.28 Drawing of One carbon pool by folate of Vibrio fischeri with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. The pathway has 18 reactions and 33 compounds. 77

5.29 Drawing of Glycine, serine and threonine metabolism of Vibrio fischeri with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. 78

5.30 Drawing of Pyruvate metabolism of Vibrio fischeri with organic (classic) layout algorithm using **VisBolic**. The grey nodes represent cofactors that have been cloned. 79

A.1 VisBolic tool logo 96

A.2 VisBolic main screen 97

List of Tables

2.1 A sample of existing pathway and enzyme databases 12

3.1 Overview of the current contents of the VisBolic database 33

3.2 The 20 compounds with the highest connectivity 37

A.1 Metabolism's elements and their attributes for querying 103

A.2 Information presented by query execution 104

A.3 Classification of metabolism's elements 105

A.4 Algorithms provided by the VisBolic tool for the analysis of metabolic pathways and networks 109

B.1 A comparison of reference features of tools for visually exploring metabolic networks 116

B.2 A comparison of functional features of tools for visually exploring metabolic networks 120

Chapter 1

Introduction

Every biological organism consists of one or more cells. The cell is the functional basic unit of all known living organisms. It was discovered by Robert Hooke in 1665. Some organisms, such as most bacteria and protozoa, are unicellular while others, such as humans, are multicellular. Inside a cell exist many interactions among the molecules (i.e. genes, gene products, metabolites). These interactions can lead to a certain product or a change in the cell, turn genes on and off or spur a cell to move.

Many important biological networks have been discovered through laboratory studies of bacteria, fruit flies, mice and other organisms. All these networks can be viewed as interconnected processes forming an intricate network of functional and physical interactions between molecular species in the cell. Many of the networks identified in these model systems are the same or have similar parts. However, it will take years of research to identify the complex connections among all the molecules in all biological networks, as well as to understand how these networks work together.

To date, at least four types of biological networks have been characterized in detail: gene regulatory networks, protein-protein interaction networks, signal transduction networks and

metabolic interaction networks. Each of these networks is discussed briefly below. This list is by no means complete and includes only these networks that are most central to research in systems biology.

- **Gene regulatory networks** contain a collection of DNA segments, which interact with each other indirectly, through their RNA and protein expression products, and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA. In general, each mRNA molecule goes on to make a specific protein (or set of proteins). In some cases this protein will be structural, and will accumulate at the cell membrane or within the cell to give it particular structural properties. In other cases the protein will be an enzyme, i.e., a micro-machine that catalyses a certain reaction. Such action is vital because genes produce proteins, which are the key components needed to carry out nearly every task in our bodies. Proteins make up our muscles and organs, help our bodies move and defend us against germs.

- **Protein-protein interaction networks** represent physical interactions of proteins with each other or with other binding partners such as DNA or RNA, often to carry out their biological function. The time scale of protein interactions ranges from very short, transient processes, to very long lasting, permanent formation of protein complexes working as molecular machines. Many of the most important molecular processes in the cell, such as DNA replication, are carried out by large molecular machines that are built from a large number of protein components.

- **Signal transduction networks** refer to any process that transports external or internal stimuli, often via signal cascades, to specific cellular parts where a cell response is triggered. Different cells are able to receive specific signals through structures on their surface, called receptors. After interacting with a receptor, the signal travels through the

cell where its message is transmitted by specialized proteins that trigger a specific action in the cell. For example, a chemical signal from outside the cell might be turned into a protein signal inside the cell. In turn, that protein signal may be converted into a signal that prompts the cell to move.

- **Metabolic Networks** are discussed in detail below.

1.1 Metabolic Networks

Metabolic pathways are series of chemical reactions occurring in a cell. Each pathway describes a step-by-step modification of an initial molecule to another. Numerous distinct pathways co-exist within a cell. This complete set of metabolic pathways is called the *metabolic network* of the cell. *Metabolism* is defined as the set of chemical reactions that happen in living organisms to maintain life. Through metabolism, which is at the core of cellular function, the energy sources that are available in the environment, are converted into Adenosine-5'-triphosphate (ATP) and other cellular building blocks, to construct essential components, such as amino acids, sugars and lipids, in order to allow organisms to grow and reproduce, maintain their structures, and respond to their environments. Metabolism is usually divided into two categories: *catabolism* and *anabolism*. Catabolism is the set of pathways that break down molecules into smaller units and release energy. Anabolism then utilizes the energy to construct essential molecules such as proteins and nucleic acids. Catabolic and anabolic pathways often work interdependently to create new biomolecules as the final end-products.

The main parts of a metabolic pathway are biochemical compounds or metabolites, enzymes, genes and reactions. *Biochemical compounds or metabolites* are small molecules that are imported/exported or synthesized/degraded inside an organism. For most metabolites, the

amount observed varies, depending on the tissue and the cell compartment inside which the compound is present. *Biochemical reactions or bio-reactions or enzymatic reactions* produce a set of one or more compounds (called *products*) from another set of one or more compounds (called *reactants or substrates*). The resulting product can be used in one of three ways:

- To be used immediately, in another subsequent reaction
- To initiate another metabolic pathway, called a flux generating step
- To be stored by the cell

A pathway is said to exist between two metabolites, if there is a connected sequence of distinct reactions between the two metabolites. Therefore, the main characteristic of metabolic pathways is that reactions are connected by their intermediates: the products of one reaction are the reactants of subsequent reactions and so on.

Although all chemical reactions are technically reversible, conditions in the cell are often such that it is thermodynamically more favourable to flow in one direction. Some reactions are *spontaneous* but most are catalysed by one or more *enzymes* (enzymatic reactions). Enzymes are proteins (i.e., macromolecules consisting of a long sequence of amino acids) or protein complexes that are coded and regulated by one or more genes and allow the cell to accelerate and modulate reaction rates. Enzymes often require dietary minerals, vitamins, and other *cofactors (or side compounds*) in order to function properly. Regulatory mechanisms, interacting with the cell's internal and external environment, are responsible for providing the cell the proper enzymes at the proper time making metabolism subject to regulation.

Metabolic pathways may either be studied in isolation or, since they are overlapping, be combined together to a metabolic network. The benefits of studying the whole network rather than individual pathways are numerous and include, for instance, the possibility to explore

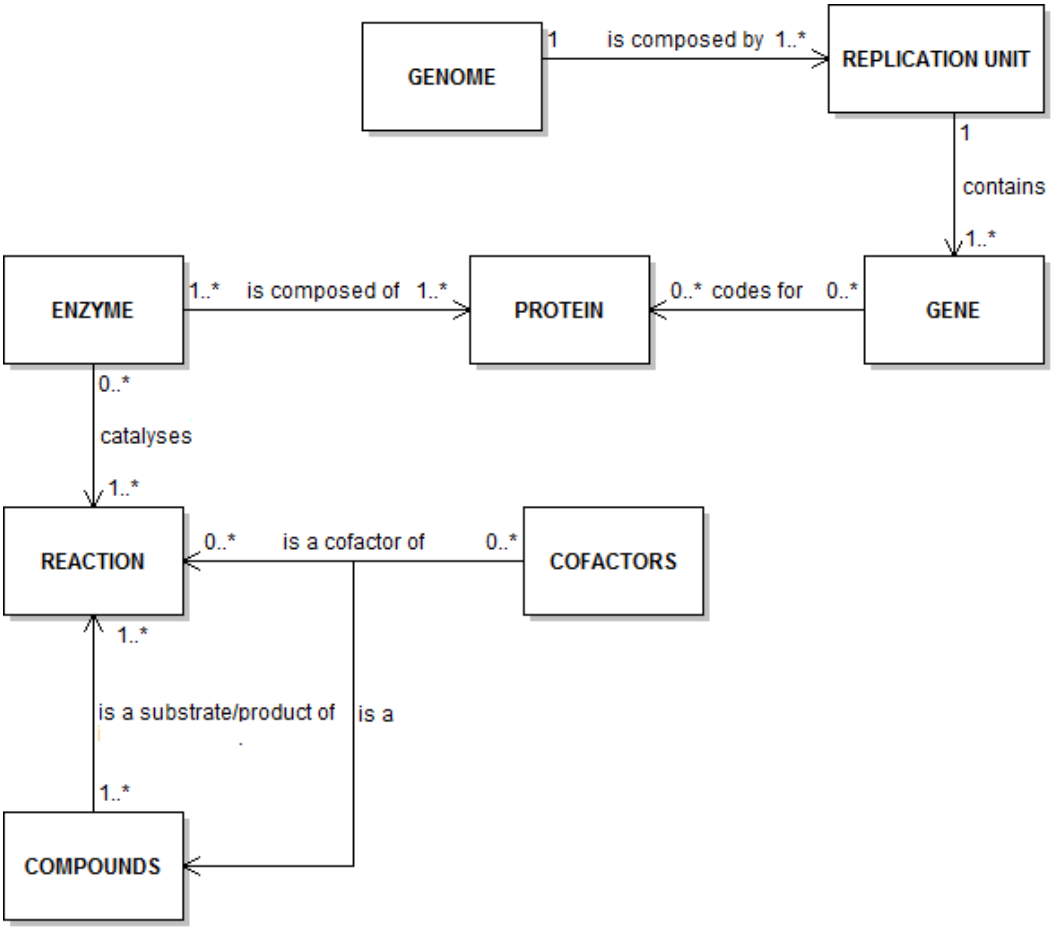


Figure 1.1: *Simplified UML diagram of the various objects involved in metabolism*

alternative pathways. The following metabolic processes [1] occur in all living organisms in some form:

- Glycolysis
- Aerobic respiration *and/or* Anaerobic respiration
- Citric acid cycle *or* Tricarboxylic acid cycle *or* Krebs cycle *or* TCA cycle
- Oxidative phosphorylation

Other metabolic pathways occurring in almost all living organisms include:

- Fatty acid oxidation *or* $\beta-$ oxidation
- Gluconeogenesis
- Amino acid metabolism
- Urea cycle *or* Nitrogen metabolism
- Nucleotide metabolism
- Glycogen synthesis *or* Glycogen storage
- Pentose phosphate pathway
- Porphyrin synthesis pathway *or* Heme synthesis pathway
- Lipogenesis
- HMG-CoA reductase pathway

[1] The term *metabolic process* refers to the organic process that is necessary for life e.g. Glycolysis, while the *metabolic pathway* concerns a specific metabolic process of a specific organism e.g. Glycolysis of E.Coli K-12

Glycolysis, Urea cycle and TCA cycle are probably the most ancient pathways. The first is discovered by Otto Meyerhof around 1940, following the work of Louis Paster and others on fermentation, and the last two by Hans Krebs in 1932 and 1937 respectively. The TCA cycle, which corresponds to the sequence of chemical reactions that produces energy in cells, is also known as the Krebs cycle and earned Hans Krebs a Nobel Prize in 1953.

Other pathways for the synthesis of energetic compounds from non-living organisms:

- Photosynthesis (plants, algae, cyanobacteria)
- Chemosynthesis (some bacteria)

It is common to observe that different organisms follow distinct pathways for the biosynthesis or degradation of the same molecule. For example, in *E.coli* methionine is synthesized in 8 steps from aspartate, whereas the yeast *Saccharomyces cerevisiae* performs this transformation in 7 steps, 5 of which are common with *E.coli* (see Figure 1.2). The case of lysine is more extreme, since *E.coli* and *S.cerevisiae* follow completely different pathways to synthesize this metabolite.

1.2 Basics in Graphs

A *graph* is a representation of the interactions among the components of a system. Examples of network components include metabolites, html documents, computers, people. More formally an *undirected* graph G=(V,E) is a mathematical structure that consists of a finite set of nodes (or vertices) V and a finite set E of *unordered* pairs of vertices, called edges. A *directed* graph G=(V,E) is a structure that consists of a finite set of nodes V (or vertices) and a finite set E of *ordered* pairs of vertices called directed edges, arcs, links or arrows.

In directed graphs, the edge (u,v) is said to be incident from u and incident to v. *Multiple* edges (also called *parallel edges* or *mutli-edges*) are two or more edges that are incident to the

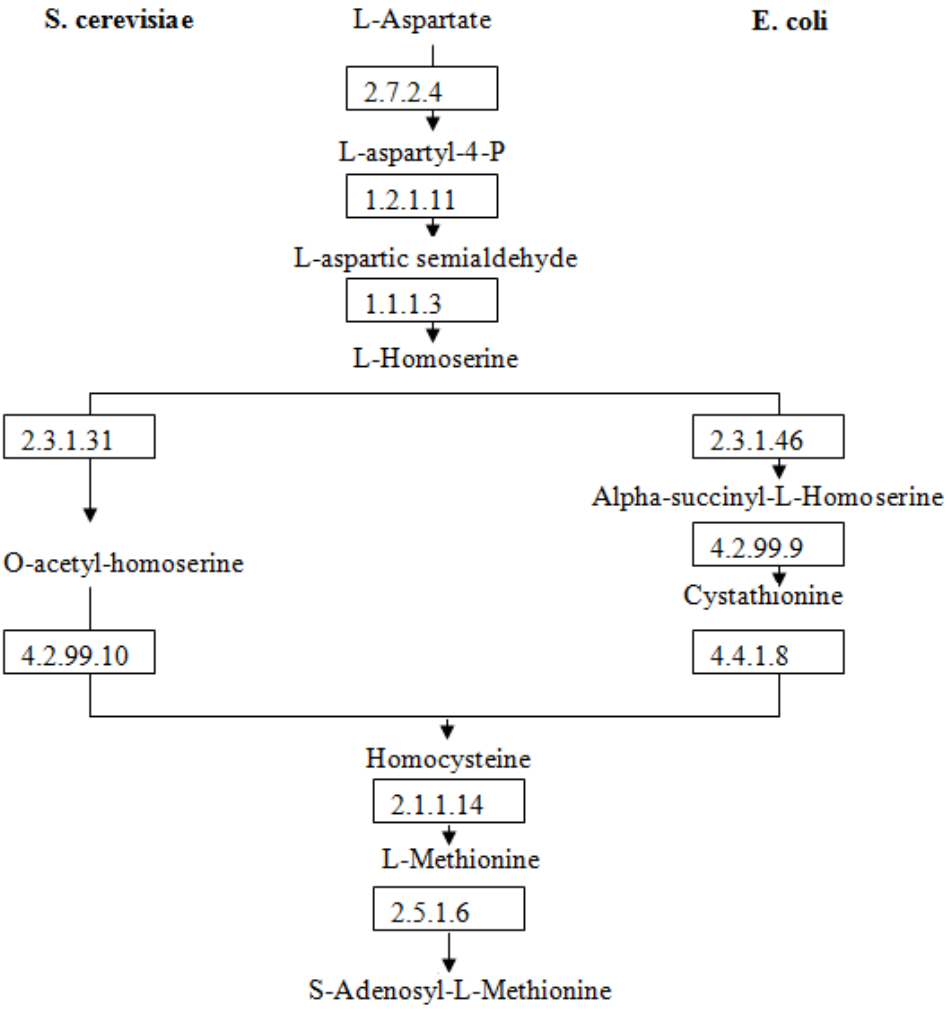


Figure 1.2: *Alternative pathways for methionine biosynthesis in E.coli and S. cerevisiae*

same two vertices. A *loop* (or *self-loop*) is an edge that connects a vertex to itself. A *simple graph* is an undirected graph, which does not contain loops or multiple edges. A *path* from a node u to a node v in a graph is an ordered sequence of distinct edges and nodes, linking a source node u to the target node v. A *cycle* is a closed path in which the source and the target nodes are the same.

From the beginning, biology has been one of the main application areas of graph theory [20]. The network represents genes, metabolites, or proteins as nodes and interaction between these molecules as edges. The directionality of a network is dependent on the characteristics of the biological data. For example protein-protein interaction networks are usually represented with an undirected network, whereas metabolic networks have directionality build into their interactions.

1.3 Motivation

Much of the data generated in the life sciences can be represented by graphs. Graph analysis and visualization methods could help in investigating cells and their biological processes. Generally, knowing the exact map of these networks and their properties is very important, since it will potentially explain the mechanisms of life processes as well as disease conditions. Such a knowledge will also serve as a key for further biomedical applications, such as development of new drugs and diagnostic approaches.

However, the areas of graph drawing and network biology are still largely disconnected. Visualization of biological networks does not apply state of the art graph drawing techniques, and graph drawing tools do not respect the drawing conventions of the life science community. Besides, little research has been performed to solve special layout and visualization problems arising in life sciences. There is a need for efficient algorithms to analyse, present and compare

biological network data. This thesis focuses on the analysis and visualization of metabolic pathways and networks, disregarding other cellular processes. A good overview on open problems and challenges in biological network visualization is provided in [10].

Chapter 2

Existing Databases and Models

Before getting to a network representation of metabolic pathways, the first task is to acquire the data. The volume of scientific data of the metabolism is exploding. Scientists will benefit from quick electronic access to information about genes, enzymes, reactions, chemical compounds and pathways, supporting their research activities. Therefore there is a need for a database that biologists can make queries and retrieve information about many aspects of metabolism, integrated with a tool for analysis and visualization of metabolic pathways. A number of metabolic databases are available electronically, some with features for querying and visualizing, with heterogeneous data formats and access mechanisms. Inconsistencies can be found by comparing database entries from different databases, e.g. regarding the names or the abbreviations used.

2.1 Existing databases

Several metabolic databases have been constructed by a number of researchers and hold data on metabolic pathways and their components (e.g. reactions, enzymes, reactants, products). Data are stored in many data repositories, with different access mechanisms, formats and

Acronym	Full Name	Scope	URL
KEGG [45, 3]	Kyoto Encyclopedia of Genes and Genomes	Genomes, enzymatic pathways and biological chemicals	http://www.genome.jp/kegg
EcoCyc [49]	Encyclopedia of Escherichia coli K-12 Genes and Metabolism	Metabolic pathways	http://ecocyc.org/
MetaCyc [49]	MetaCyc - Encyclopedia of Metabolic Pathways	Metabolic pathways	http://metacyc.org/
ERGO (ex WIT)	What Is There; Enzymes and Metabolic Pathways	Enzyme function, metabolic pathways, metabolic reconstruction	http://ergo.integratedgenomics.com
aMAZE [55]	Database of protein function and biochemical processes	Metabolic pathways, regulation and signal transduction	http://www.ebi.ac.uk/research/pfbp
ExPASy [36]	Expert Protein Analysis System	Metabolic Pathways (Electronic form of Boehringer-Mannheim map)	http://www.expasy.ch/cgi-bin/search-biochem-index
BRENDA [66]	The Comprehensive Enzyme Information System	Enzyme Function	http://www.brenda-enzymes.info/
EMP [16]	Enzymes and Metabolic Pathways	Enzyme Function	http://wit.mcs.anl.gov/WIT2/
Biocatalysis-Biodegradation [30]	The University of Minnesota biocatalysis/biodegradation database	Metabolic pathways	http://www.labmed.umn.edu/umbbd/index.html

Table 2.1: *A sample of existing pathway and enzyme databases*

structures. Pathway databases are closely related to enzyme databases. Enzyme databases mainly contain information about enzymes and their properties. There are several databases on metabolic pathways and enzymes [80] (see table 2.1). These databases contain hundreds of metabolic pathways and thousands of chemical reactions. Even the metabolic pathway for a small organism constitutes a large network e.g. the metabolic pathway for bacterium *E.Coli K-12 DH10B* consists of 982 compounds linked by 1064 reactions!

2.1.1 The KEGG database

The KEGG database contains, among others, all known metabolic pathways and consists of three main databases (related to metabolic pathway information) which are tightly connected:

- Genes: contains organism-related genome and gene information and lists of genes within an organism and pathway.
- Ligand: contains information about compounds, enzymes and reactions.
- Pathway: contains the graphical representations of all known pathways and lists of enzymes and reaction within pathways (see Figure 2.1).

KEGG also contains DBGET database retrieval system. The query interface of KEGG does not allow complex queries e.g. a query for all known reactions that use a defined substrate. Graphical representations are drawn manually, are non - dynamic and consists of links to specific information about compound, enzymes, reactions and genes. Reactions in pathway maps do not represent cofactors e.g. ATP or NADH. Pathways maps are also linked to other related pathways, connected by their common compounds. This gives an overview of connections to other pathways, e.g., one main final product of *Glycolysis* is used in the pathway of *Amino acid metabolism* [45, 3, 39]. A comparison of pathways is not implemented.

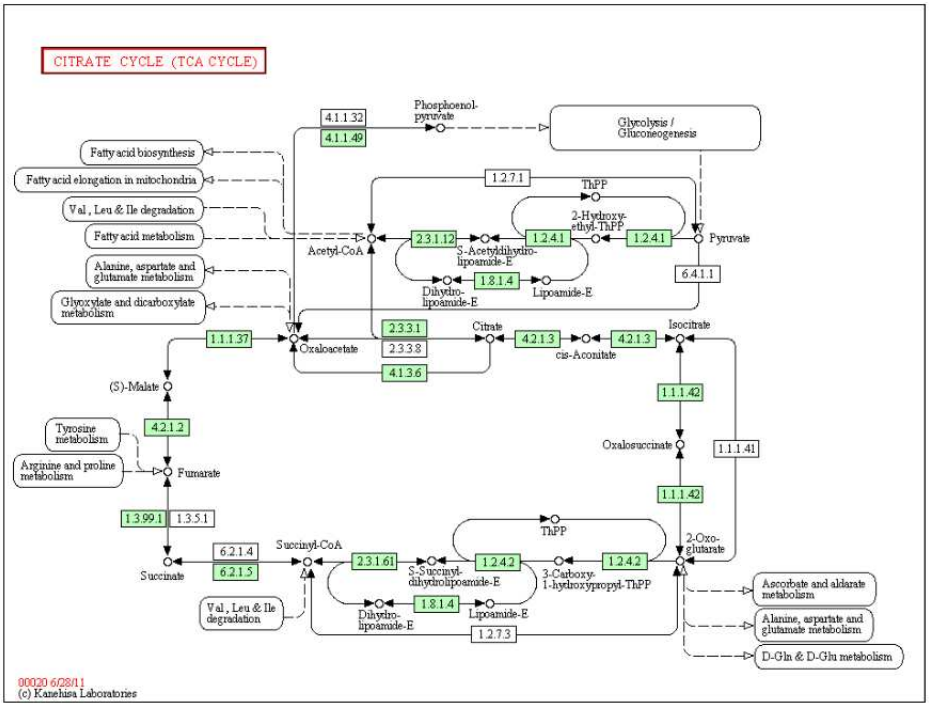


Figure 2.1: *The TCA cycle or Krebs cycle or Citrate cycle of the bacterium Escherichia coli K-12 DH10B as presented in KEGG database. The circles represent metabolites whereas the green boxes depict the enzymes that catalyse the corresponding reactions.*

2.1.2 EC Numbers

The "Enzyme Commission" (EC) numbering system is used to classify enzymes by type and function, by means of a hierarchical code with four digits separated by dots, attributed by the International Union of Biochemistry and Molecular Biology (IUBMB). Each digit represents a progressively finer class [77]. The first one defines the type of reaction catalysed:

1. oxidoreductases
2. transferases
3. hydrolases
4. lyases
5. isomerases
6. ligases

The next two further indicate, respectively, the subclass and sub-subclass of the reaction, and the fourth is the serial number of the enzyme in its sub-subclass (see [6] for more details).

2.1.3 Summary

A comparison of data included in pathway databases is complicated because of different classifications of compounds, genes, proteins, pathways and gene/protein functions. Only the classification of enzymes recommended by the International Union of Biochemistry and Molecular Biology (IUBMB) is used as a standard. Inconsistencies arise also from names and abbreviations used in different databases. The above inconsistency is transferred to inconsistent classifications of pathways (2.2).

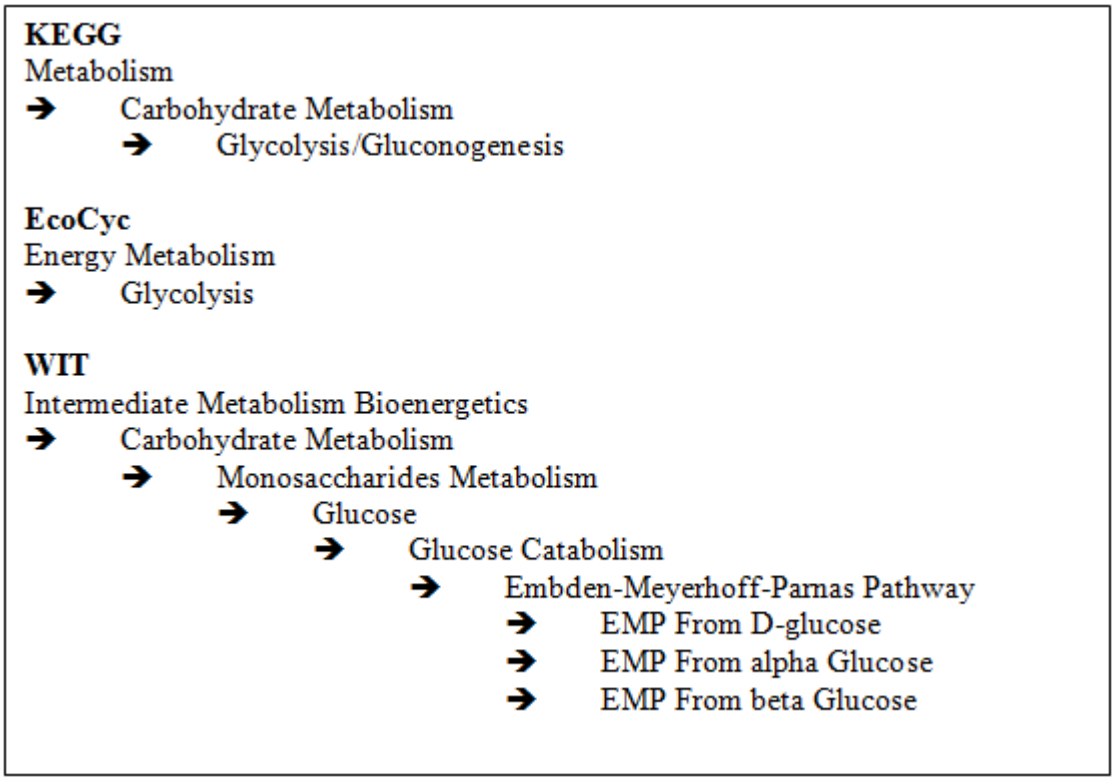


Figure 2.2: *Classification of Glycolysis in KEGG, EcoCyc and WIT database. The figure is borrowed from [80].*

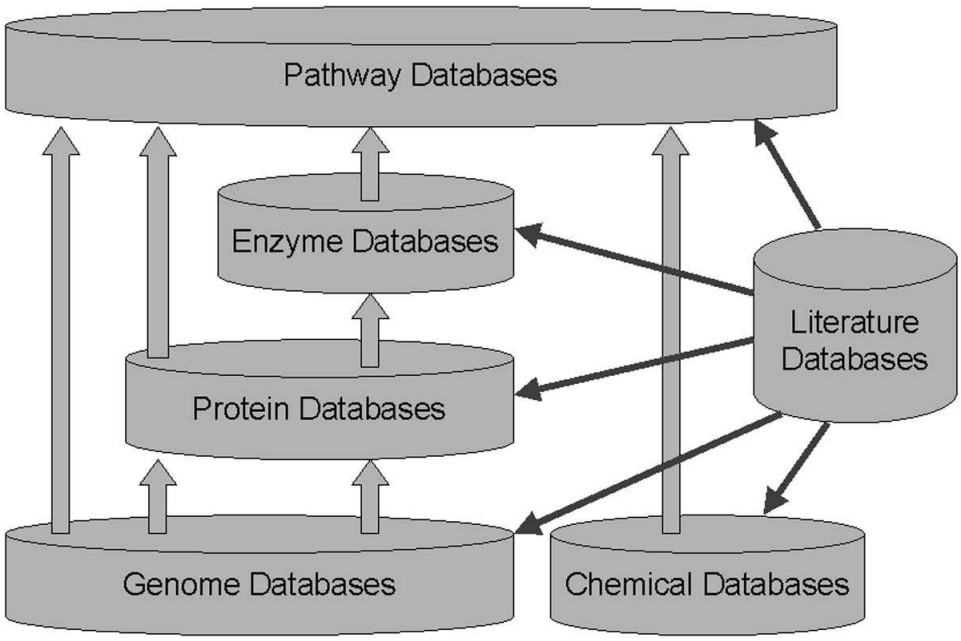


Figure 2.3: *The connectivity of the main pathway database types. The figure is borrowed from [80].*

Genome and chemical databases can be seen as the basis for all other databases. Pathway databases are located above other databases, containing parts of data from the databases below, as you can see in Figure 2.3. Consequently pathway databases might also contain the errors made in other databases.

In terms of the amount of data incorporated in the database, the KEGG system clearly deserves the highest praise. However users familiar with the biochemical pathways of E.Coli, the EcoCyc system offers the most information on that specific organism.

2.2 Existing Graph-based models

Distinct models are used for the dynamic and structural study of metabolism, the most common being graphs. Various forms of graph - based models have been devised and used for the analysis and visualization of metabolic pathways. Each model has different objectives and provides its

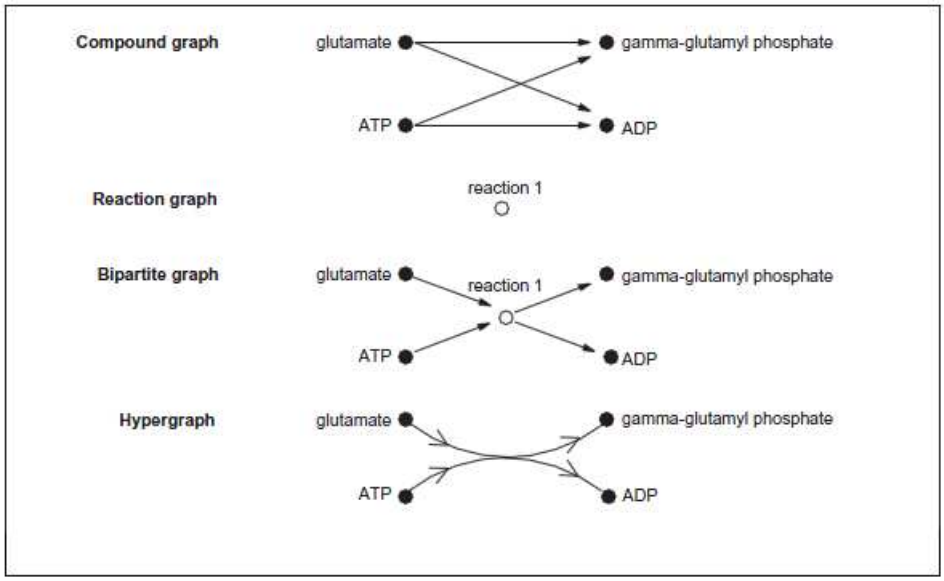


Figure 2.4: *Reaction:* $glutamate + ATP \longrightarrow gamma - glutamylphosphate + ADP$. *The figure is borrowed from [24].*

own view of reality, taking into account particular aspects while neglecting others [24].

2.2.1 Compound or Substrate Graphs

In compound graphs the nodes are the chemical compounds. A directed edge connects compound A to compound B if A occurs as a reactant and B as product in the same reaction. For example the compound graph for the following reaction is shown in Figure 2.4:

$$glutamate + ATP \longrightarrow gamma - glutamylphosphate + ADP$$

Compound graphs are usually non-planar. For example, a reaction with 3 reactants and 3 products corresponds to $K_{3,3}$ graph. Therefore, according to *Kuratowski theorem* [4] the corresponding compound graph is non-planar. Also the descriptive power of compound graphs is poor, since the structure of the reaction is lost in them: one can no longer distinguish if two

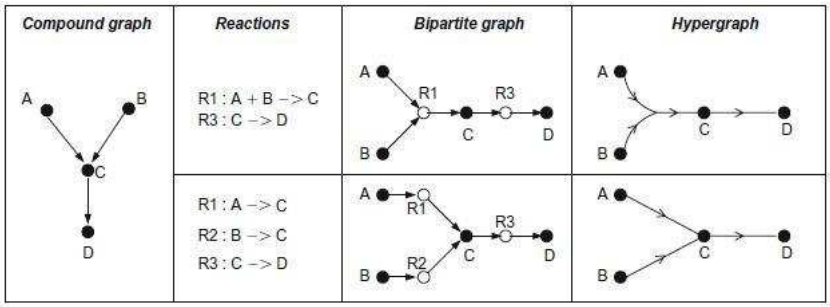


Figure 2.5: *Ambiguity in existing models. The figure is borrowed from [24].*

compounds are involved in the same reaction. As a consequence different sets of reactions can lead to the same compound graph, as shown in Figure 2.5:

2.2.2 Reaction Graphs

In reaction graphs, the nodes represent the reactions. An edge connects two reactions R1 and R2, if a compound is both a product of R1 and reactant of R2, i.e. if they share the same compound. The edges can be directed or undirected, depending on whether the reactions are reversible or not. In 2.4 the reaction graph has a single node as it only involves one reaction. Reaction graphs are very limited since compounds and enzymes are not represented. Also they are ambiguous since different set of reactions can also lead to the same reaction graph.

2.2.3 Enzyme Graphs

In enzyme graphs the nodes represent the enzymes. An edge exists between two enzymes, if they catalyse reactions that share a compound. Using enzyme graphs may lead to confusion about the structure of the network, i.e. reactions catalysed by the same enzymes are merged, creating short-cuts between distant compounds. Nevertheless, this model may still be adopted, if the focus is really on enzymes and on the relations between them.

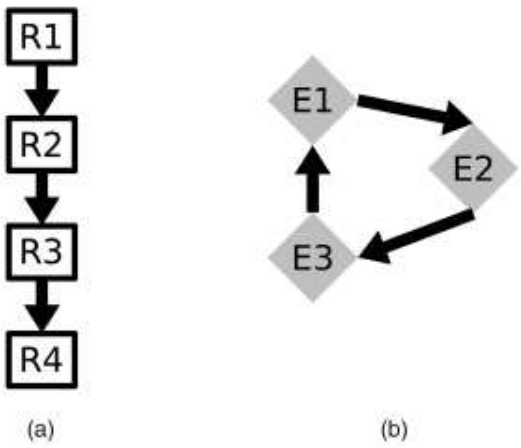


Figure 2.6: *Reaction and Enzyme graph for the sets of reactions: R1: A + B → C + D, R2: D + E → F + G, R3: F + G → H + I, and R4: I → J + K. E1, E2 and E3 are enzymes: E1 catalyses R1 and R4, E2 catalyses R2 and E3 catalyses R3. (a) Reaction graph (b) Enzyme graph*

Compound, reaction and enzyme graphs give limited information about the structure of the system, while they are ambiguous. More expressive models are *bipartite* and *hypergraphs*.

2.2.4 Bipartite and Hypergraphs

In bipartite graphs there are two set of nodes, one set corresponds to metabolites and the other to reactions. No edges can relate nodes from the same set. A directed edge from a compound to a reaction node denotes reactant, while an edge from a reaction to a compound node denotes a product of the reaction (see Figure 2.4).

Another way of solving the ambiguity of simple graphs is to use hypergraphs [11, 35, 13, 12, 74, 75]. Hypergraphs have been used to model many-to-one, one-to-many, and many-to-many relationships between objects. The directed hypergraphs have also appeared with different names including "Labelled graphs" and "And-Or" graphs. The edges in a hypergraph, known as hyperedges or hyperarcs, can relate groups of more than two nodes. Since a reaction may have

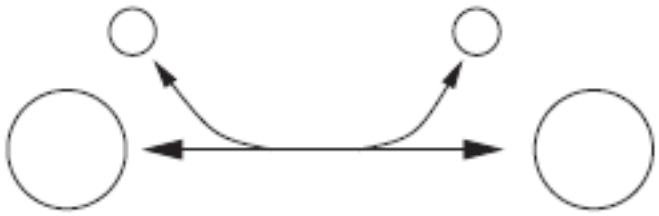


Figure 2.7: *Metabolic reaction as given in many biochemical textbooks. This kind of edge is called hyperedge.*

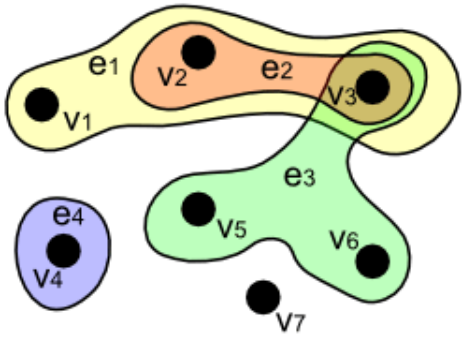


Figure 2.8: *Graphical representation of an undirected hypergraph with V = { v1, v2, v3, v4, v5, v6, v7 }, E = { e1, e2, e3, e4 } = { { v1, v2, v3 } , { v2, v3 } , { v3, v5, v7 } , { v4 } }. Source:* ***http://en.wikipedia.org/wiki/Hypergraph.***

more than one reactant or more than one product, the pathway is better represented by directed hypergraphs, where hyperedges represent reactions and nodes represent metabolites. In any case, formally, metabolic networks *are directed hypergraphs*. Moreover, the usual representations of metabolic reactions in biochemical textbooks use hyperedges as given in Figure 2.7.

An *undirected hypergraph* H is a pair of finite sets (V,E_h), where V = $\{v_1, v_2, ..., v_n\}$ is the set of nodes and $E_h = \{e_{h1}, e_{h2}, ..., e_{hm}\}$ with $e_{hi} \subseteq$ V, for i=1,2,....m is the set of hyperedges. A hyperedge can be seen as a n-ary relation between nodes and can be normally written in set notation such as { u, v, w } (see Figure 2.8).

A *directed hypergraph* H is a pair of finite sets (V,E_h) where V is the set of vertices as before and the elements of E_h are ordered pairs of non-empty subsets of V called *directed hyperedges*. A directed hyperedge is written as $e_h = \langle$ T,X $\rangle$, where T $\subseteq$ V is the *tail* of e_h, and X $\subseteq$ V is the *head* of e_h. A *hyperpath* is defined as a sequence $P = \{u_1, e_1, u_2, e_2, ..., e_{n-1}, u_n\}$, where $\forall$ i, $1 \leq i \leq n-1 : u_i$ is a hypernode, e_i is a hyperedge, $u_i = Tail(e_i)$ and $u_{i+1} = Head(e_i)$. A hyperpath is a hypercycle if $u_n \bigcap u_1 \neq \emptyset$.

Directed hypergraphs have been used for modelling problems from diverse application domains such as propositional logic, relational databases [35], secret transfer protocol [75]. In [52] the need of using hypergraphs in modelling cellular networks has been addressed. Existing pathway analysis is mostly performed on simplified models, such as compound, reaction or bipartite graphs that permit the use of existing graph based algorithms.

Hypergraphs are often projected onto graphs, losing some information but making their drawing easier and their analysis amenable to the huge corpus of methods and algorithms from graph theory. Algorithms for hypergraphs will differ in complexity from similar algorithms for graphs. Since graphs are special cases of hypergraphs, algorithms for hypergraphs are at least as hard as its specialized implementations in the graph case. On the other hand problems that are computationally easy for graphs can be hard for hypergraphs.

Chapter 3

Proposed Model and Database

In some cases the use of simple and bipartite graphs to represent complex networks, does not provide a complete description of the real-world systems under investigation. Various problems arise from adopting these models for representing the metabolic knowledge, as presented below. It should be noticed that reaction and enzyme graphs are ignored from the beginning, since they do not represent all the information of the metabolism.

3.1 Why not compound graphs?

In a collaboration network represented by a simple graph (i.e. a graph of author nodes and co-author links), we only know whether scientists have collaborated or not, but we are not able to know whether three or more authors, that linked together in the network, were co-authors of the same paper or not. Similarly, metabolic networks are often modelled as compound graphs, where nodes represent metabolites and edges represent reactions, relating the metabolites involved in the reaction. This representation is often not fully correct and various problems may arise:

- Many functional relationships in metabolic networks are more complicated than what can be represented in compound graphs, i.e. a reaction may have more than one reactant or more than one product, a scenario that can not be captured by ordinary graphs. For example a metabolic reaction such as $A+B \longrightarrow C+D+E$. Such multilateral relationships are not compatible with graph edges.

- By treating each metabolite in a reaction separately, dependencies between the metabolites are lost.

- There may be more than one reaction involving any two metabolites. Ordinary graphs are not ideal for representing multiple connections between metabolites.

- Reactions can not be reconstructed from their graphical representation.

Although it is possible to represent metabolic networks using simple graphical representation, it often leads to information loss and wrong computation of network properties. For example, the computation of k-core [1] for the metabolic network in Figure 3.1 gives a 2-core consisting of { A, C, E } in contrast to the graph model, whose maximum k-core is a 3-core consisting of the nodes { A, B, C, D }, i.e weighting B and D stronger than E.

A more plausible and accurate modelling framework can be offered by hypergraphs. Hypergraphs are a generalization of ordinary graphs, where a hyperedge, can connect any number of vertices. The vertices are the compounds and the hyperedges are the reactions, connecting the compounds. This makes them ideal for representing the relationships between multiple metabolites in a reaction. Implementing hypergraphs to model different network biology analysis problems has several advantages over traditional graph models, avoiding all problems mentioned above.

[1] k-core is the maximal node-induced sub-graph in which all nodes have a degree equal or larger than k

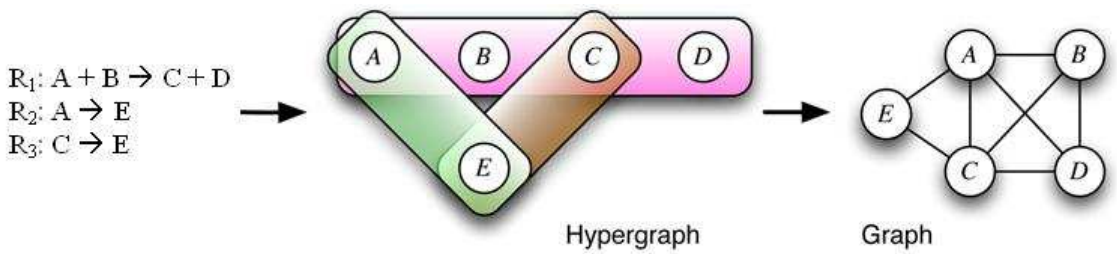


Figure 3.1: *Example of undirected hypergraph arising in the context of biological network analysis. The figure is borrowed from [52].*

On the other hand, unlike graphs, visualization of hypergraphs is difficult and often network analysis using hypergraphs is more theoretical than pictorial. Although hypergraphs occur ubiquitously when dealing with cellular networks their notion is known to a much lesser extent than that of graphs.

3.2 Why not bipartite graphs?

A metabolic pathway in Figure 3.2 has been drawn as: directed hypergraph, directed substrate graph and directed bipartite graph. In the latter, reactions and compounds are represented as two different types of nodes. Edges exist only from metabolites to reactions and vice versa, representing substrate-reaction and reaction-product relationships respectively. In contrast to the compound graph, bipartite graph still reflects the original information and can be used to determine a number of topological properties using graph-theoretical techniques.

However graph theoretic methods may not be appropriate when analysing functional properties that require a consideration of the AND connections between reactants and products. Besides, the homogeneity in the definition of nodes is lost, since there are two classes of nodes with completely different interpretations. This may lead to artifacts in the data, in the study of various topological properties.

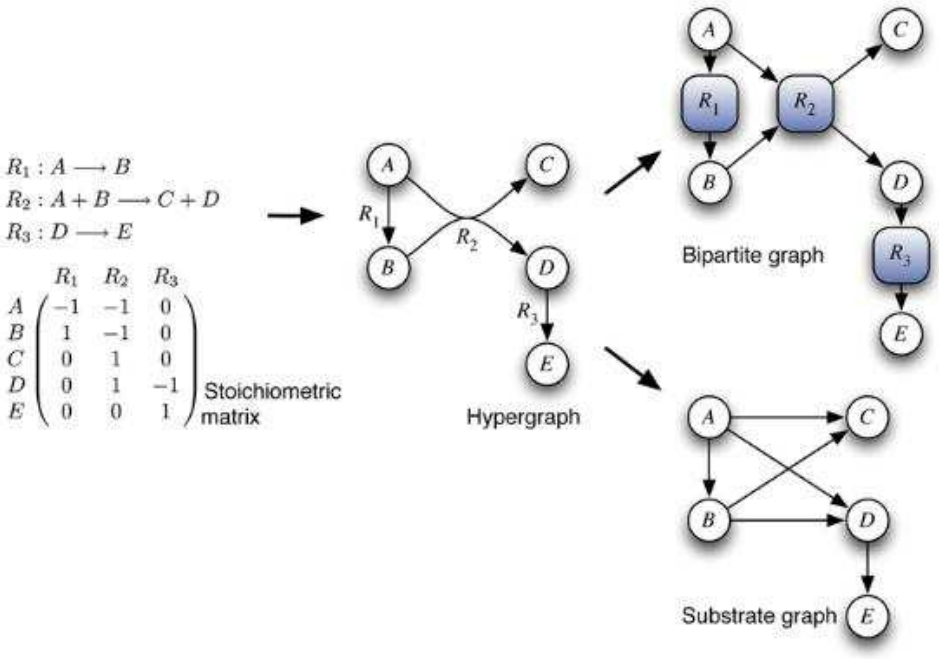


Figure 3.2: *Example of directed hypergraph arising in the context of biological network analysis. The figure is borrowed from [52].*

For example in Figure 3.2, the execution of reaction R2 demands the presence of compounds A *and* B. However in a bipartite graph a path from A to E exists via nodes A, R2, D and R3, that may suggest that this is possible! A natural way of representing these systems is to use hypergraphs.

3.3 Proposed Model

The hypergraph model that will be used for the representation of metabolic knowledge is the following: A directed hypergraph H is a pair of finite sets (V,E_h) where V is the set of vertices and E_h is the set of hyperedges. A directed hyperedge is written as $e_{h1} = \langle$ T,X $\rangle$, where T,X $\in$ P(V) $\setminus\{\emptyset, V\}$. The tail of a hyperedge corresponds to the **main** reactants and the head to the **main** products of the reaction represented by the hyperedge. The head and tail of a hyperedge can not be the empty set, since every reaction has at least one main reactant and at least one main product.

In order to be able to use existing graph-based techniques for the analysis of metabolic pathways and networks, the model is extended. We consider the tail and head of a hyperedge as two *single entities*. For that we define the notion of a *hypernode*. The set of hypernodes V_h is defined as: $V_h = \bigcup_{i=1}^{m}\{\{Tail(e_{hi})\}, \{Head(e_{hi})\}\}$, where $e_{h1}, e_{h2}, ..., e_{hm}$ are the hyperedges of H, Tail(e_{hi}) is the tail of the hyperedge e_{hi} and Head(e_{hi}) is the head of the hyperedge e_{hi}. We also assume that there is an edge, called *pseudo-edge*, from hypernode u_{h1} to hypernode u_{h2} and vice versa if $u_{h1} \bigcap u_{h2} \neq \emptyset$. We regard that the length of a pseudo-edge is 1. The set of pseudo-edges is denoted by E_p.

In Figure 3.3, the set of reactions is { r1, r2, r3, r4, r5, r6, r7 }, the set of compounds is { A, B, C, D, E, F, G, H, I }, and the set of hypernodes is { {A}, {B}, {D,E}, {B,C}, {D,E,F}, {G}, {C}, {F}, {H,I} }. Indicatively, since $\{D, E\} \bigcap \{D, E, F\} \neq \emptyset$, we assume that there is a

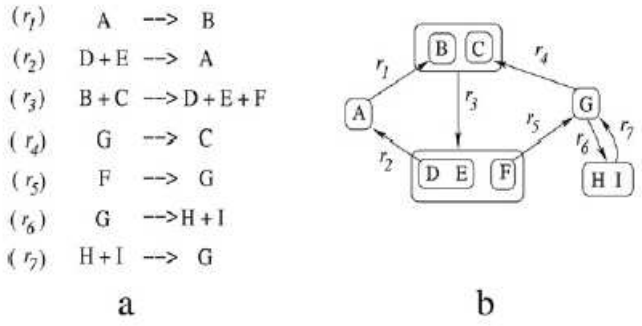


Figure 3.3: *(a) Example of reactions and (b) their directed graph representation (introducing the notion of hypernodes). The figure is taken from [60].*

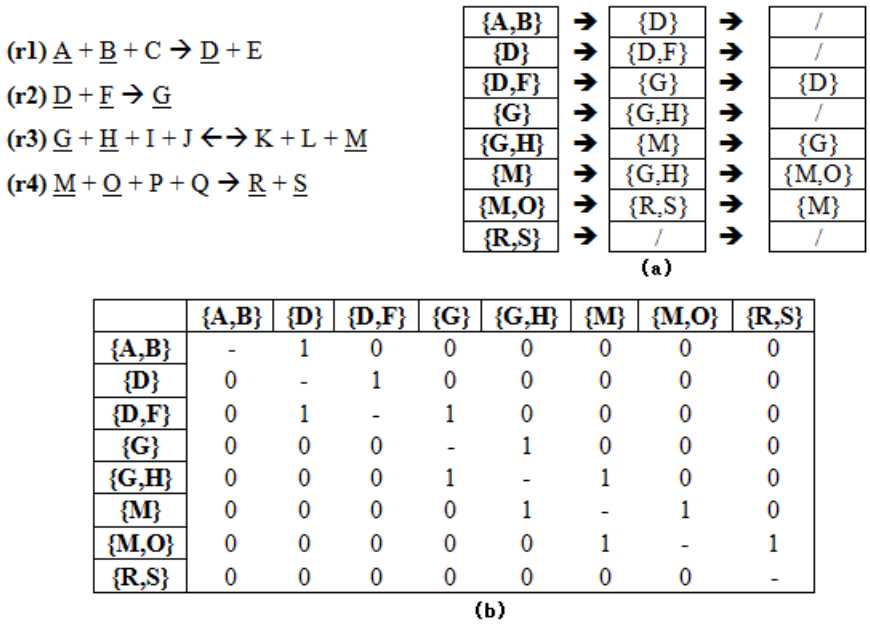
pseudo-edge from $\{D, E\}$ to $\{D, E, F\}$ and vice versa.

For the representation of metabolic knowledge, two options are available: adjacency list or adjacency matrix representation. Since networks [2], are sparse, adjacency list representation is chosen. However adjacency matrix representation is required for the execution of some algorithms, such as the all-pairs shortest paths algorithm. In Figure 3.4, a simple, hypothetical metabolic pathway is presented and its respective adjacency list and adjacency matrix representation.

3.4 Hypergraph Drawings

The two most common methods to draw hypergraphs are the subset-based and the edge-based method. A subset-based drawing highlights the fact that a hypergraph can be interpreted as a set system: Vertices are drawn as points in the plane and hyperedges are drawn as simple closed curves that contain exactly those vertices that they are incident to (see Figure 3.5(a)). It is easy to see that any hypergraph can be drawn this way. Subset-based drawings neither

[2]The term *network* refers to a metabolic pathway or a metabolic network



	{A,B}	{D}	{D,F}	{G}	{G,H}	{M}	{M,O}	{R,S}
{A,B}	-	1	0	0	0	0	0	0
{D}	0	-	1	0	0	0	0	0
{D,F}	0	1	-	1	0	0	0	0
{G}	0	0	0	-	1	0	0	0
{G,H}	0	0	0	1	-	1	0	0
{M}	0	0	0	0	1	-	1	0
{M,O}	0	0	0	0	0	1	-	1
{R,S}	0	0	0	0	0	0	0	-

Figure 3.4: *Example of a simple, hypothetical metabolic pathway. (a) Its adjacency list representation; (b) its adjacency matrix representation. Underlined metabolites determine main compounds.*

know any concept of or experience any problems with planarity. Bertault and Eades [18] show how to create subset-based hypergraph drawings.

An edge-based hypergraph drawing resembles standard drawings of graphs more closely [56]. Vertices are again drawn as points, but hyperedges are drawn as Steiner trees (see Figure 3.5(b)). Edge-based drawings imply the most common definition for hypergraph planarity: A hypergraph is planar if and only if it has an edge-based drawing without hyperedge crossings.

Another way to draw hypergraphs is the so-called Zykov representation. Vertices are again drawn as points, but hyperedges are visualized by faces of a subdivision. The vertices around a face of the subdivision are the ones connected by the corresponding hyperedge (see Figure 3.5(c)). To distinguish faces that represent hyperedges from faces that do not, a background color is needed to fill all faces of the subdivision that do not correspond to hyperedges. A hypergraph is Zykov-planar if it has a Zykov representation. Zykov-planarity is equivalent to

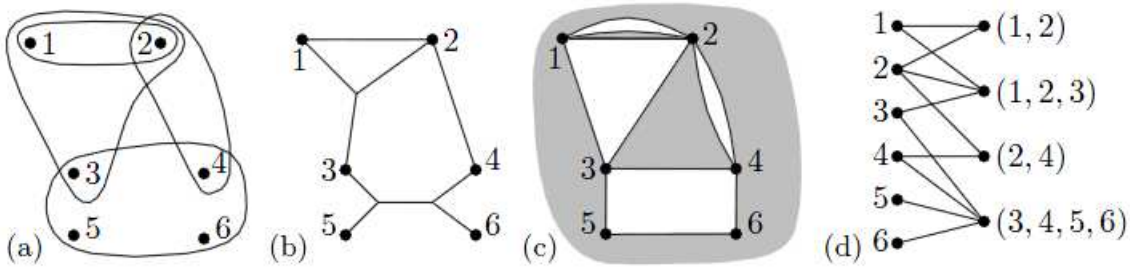


Figure 3.5: *Four drawings of the hypergraph H = (V,E) with* $V = \{1, 2, 3, 4, 5, 6\}$ *and* $E = \{\{1, 2\}, \{1, 2, 3\}, \{2, 4\}, \{3, 4, 5, 6\}\}$*: (a) subset-based drawing; (b) edge-based drawing; (c) Zykov representation; (d) bipartite graph. The figure is borrowed from [50].*

hypergraph planarity as induced by edge-based drawings.

A hypergraph H can also be drawn as a bipartite graph, where one set of vertices corresponds to the vertices of H and the other set corresponds to the hyperedges (see Figure 3.5(d)). The edges of the bipartite graph represent vertex-hyperedge incidences. Also this representation immediately implies a definition of planarity, which is equivalent to Zykov-planarity and hence to hypergraph planarity as induced by edge-based drawings [78].

Many hypergraphs are not (Zykov-) planar and hence have neither an edge-based drawing, nor a Zykov representation, nor a drawing as a planar bipartite incidence graph. Motivated by this fact, Pollak and Johnson study alternative definitions of hypergraph planarity which are implied by yet two more methods to draw hypergraphs: hyperedge based Venn diagrams and vertex based Venn diagrams (The names are somewhat unfortunate since the drawings have little in common with standard Venn diagrams).

In a hyperedge based Venn diagram, hyperedges correspond uniquely to the faces of a planar subdivision in such a way that for any vertex v, the union of the faces corresponding to hyperedges that contain v, is connected. In a vertex based Venn diagram, vertices correspond

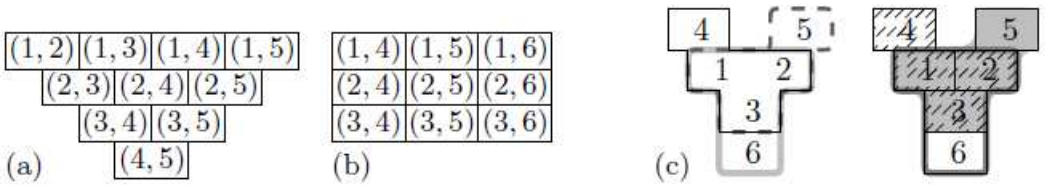


Figure 3.6: *A hyperedge based Venn diagram for (a) K_5; (b) $K_{3,3}$; (c) two different drawings of a vertex based Venn diagram for H with $V = \{1,2,3,4,5,6\}$ and $E = \{\{1,2,3,4\},\{1,2,3,5\},\{1,2,3,6\}\}$. The figure is borrowed from [50].*

uniquely to faces of a planar subdivision in such a way that for any hyperedge the union of the faces corresponding to its incident vertices is connected. Note that the subdivision itself does not show the hyperedges directly, curves that enclose unions of faces must still be drawn to visualize them. A hypergraph is hyperedge planar or vertex-planar if a planar subdivision with the required properties exists. It is easy to see that both K_5 and $K_{3,3}$ are hyperedge-planar (see Figure 3.6(a,b)). Concerning vertex-planarity, consider the hypergraph H in 3.6(c). H is vertex-planar but not (Zykov-)planar. Figure shows two drawings of H which showcase different methods to draw the hyperedges in a vertex-based Venn diagram.

The drawing that is preferable for the representation of metabolic pathways and networks is the edge-based drawing. Edge-based drawing is the only for representing the directionality of the reactions (except bipartite graphs, whose weaknesses are mentioned above).

3.5 VisBolic Database

The data of VisBolic Database are derived from KEGG database. The main reasons behind selecting KEGG as the data source for VisBolic database were the versatility of data present in KEGG, its completeness and the fact that the KEGG data are updated on a regular basis.

```
ENTRY       EC 1.11.1.10
NAME        Chloride peroxidase
CLASS       Oxidoreductases
            Acting on a peroxide as acceptor
SYSNAME     Chloride:hydrogen-peroxide oxidoreductase
REACTION    2 RH + 2 Cl- + H2O2 = 2 RCl + 2 H2O
SUBSTRATE   RH
            Cl-
            Br-
            I-
            H2O2
PRODUCT     RCl
            RBr
            RI
            H2O
COFACTOR    Heme
COMMENT     Brings about the chlorination of a range of
            organic molecules, forming stable C-Cl bonds.
            Can also act on Br- and I-. Probably a heme-
            thiolate protein (P-450).
STRUCTURES  PDB: 1CPO 2CPO 1VNC
```

Figure 3.7: *Example entry from KEGG*

The VisBolic database contains, among others, information about genes, enzymes, reactions, chemical compounds and a description of metabolic pathways for a plethora of organisms. The KEGG data are stored in flat files (see Figure 3.7), making the development of a parser necessary for storing them in a relational database, making them amenable to computer analysis. It should be noticed that the flat files for metabolism have 5,38 GB size.

The database has been implemented using mySQL relational database system. By using mySQL the biologists can use available user interfaces for introducing and handling the data. Another reason for having a relational database is related to performance. With the amount of the data that are currently handling (over 7 million entries), the database performs well.

The VisBolic database does not contain more information about genes, because that would be beyond the scope of this thesis, since we are focused on a graph and hypergraph theoretic approach of metabolic networks. Besides, concentrating information about genes (which already are too many, 909405 genes!) will overload the database and would decrease its efficiency, given that the resources are limited to the use of a common personal computer.

Non-redundancy is a basic principle of database design and underlies the normal forms of

Objects in the database	Number of objects stored
Chemical Compounds	16788
Reactions	8389
Enzymes	5246
Genes	909405
Processses	272
Organisms	1437
Pathways	94808

Table 3.1: *Overview of the current contents of the VisBolic database*

database theory. A database with minimal redundancy should take less time to acquire, will be easier to update when errors are detected (since information must be updated only in one place) and will not accumulate errors due to the fact that updates will require the change only one element from a redundant set. VisBolic database is in Third Normal Form (3NF). The EER model of the VisBolic database is presented in Figure 3.8.

Finally, it should be noticed that KEGG database contains some inconsistencies. It was attempted to remove them during the creation of VisBolic database or to "catch" them during execution of VisBolic application, but may even exist some, causing "abnormal" execution of VisBolic. For example, although every reaction should have at least one main reactant and at least one main product, there are reactions that do not have any reactant or any product. For example, the reaction R02869 in the *Arginine and proline metabolism* of *Escherichia coli C ATCC 8739* organism, does not have any main reactant and product.

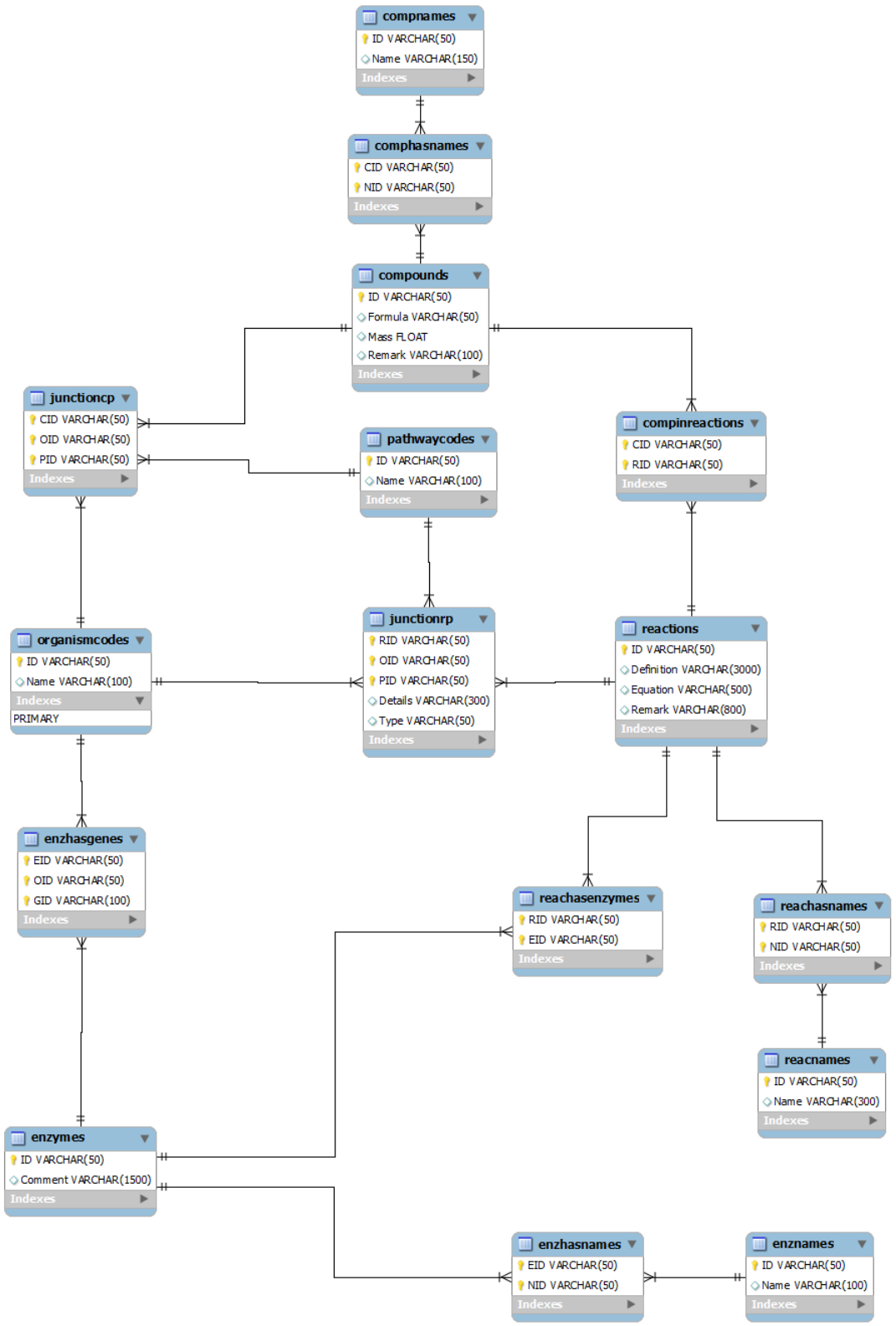


Figure 3.8: *EER Model of the VisBolic database containing its main relations (diagram created with mySQL Workbench)*

3.6 Treatment of cofactors in analysis

In a given reaction some compounds are considered as primary (main) and others as auxiliary (cofactors). Usually the connectivity of cofactors is very high because they behave as energy source and reducing agent in many reactions, while the vast majority of compounds is involved in a very limited number of reactions. Considering them as regular compounds to build a model, would lead to an artificially large number of links and to misleading conclusions when computing network measures. This improper treatment of cofactors may produce acceptable solutions from the mathematical point of view, but these solutions will be devoid of biochemical meaning. In table 3.2 compounds with the highest connectivities are presented.

A classical way of facing this problem is to ignore all cofactors from the network, for analysing it, in order to produce biologically meaningful results. This approach is followed here. The ignoring of all cofactors has several caveats:

1. One needs to heuristically define a threshold to determine how many metabolites to remove.

2. Some highly connected metabolites, like pyruvate, will be removed, whereas it is commonly agreed that they belong to the main backbone of metabolism.

3. Even compounds, like ATP, that could be safely removed from most reactions, should not be eliminated from those that participate in their own synthesis.

Hence, a compound can not be characterized as a cofactor for every reaction that participates in. This characterization depends on the reaction, on the process and on the organism. Therefore, the successful distinction between main compounds and cofactors for each reaction is not always available. It should be noticed that in a valid pathway, if a metabolite is acting as

a main compound in any of the reactions of the pathway, then it cannot be used as a cofactor in the same pathway.

In [48], Karp and Paley introduce a heuristic method for inferring main compounds and cofactors. Although KEGG maps represent only main compounds, KEGG database contains knowledge about side compounds for each reaction in each organism. That knowledge will be used, for distinguishing between main compounds and cofactors.

Compound ID	Name	Substrate	Product	Sum
C00001	H2O Water	1311	802	2113
C00080	H+ Hydron	489	820	1309
C00007	Oxygen O2	784	13	797
C00006	NADP, NADP+	382	373	755
C00005	NADPH	372	381	753
C00002	ATP	549	21	570
C00003	NAD+	376	91	467
C00004	NADH	86	372	458
C00011	CO2, Carbon Dioxide	34	416	450
C00009	Orthophosphate	96	336	432
C00008	ADP	20	387	407
C00010	CoA, Coenzyme A	163	234	397
C00013	Diphosphate	38	328	366
C00014	NH3, Ammonia	23	257	280
C00025	L-Glutamate	63	129	192
C00019	S-Adenosyl-L-methionine	178	2	180
C00020	AMP	17	162	179
C00026	Oxoglutaric acid	134	45	179
C00022	Pyruvate	64	102	166
C00021	S-Adenosyl-L-homocysteine	6	158	164

Table 3.2: *The 20 compounds with the highest connectivity*

Chapter 4

Analysis of Metabolic Pathways and Networks

In recent years, the development of novel techniques have led to the generation of huge data sets in the life sciences. Manual data analysis is costly and time-consuming. One way to cope with the enormous amount of the data, is the computational methods that support data analysis. Pathway analysis is hence a challenging problem in bio-informatics and systems biology.

Interaction data can be represented with networks, whose topological structure contains significant biological properties. Several of the most important and commonly used topological measures include degree distribution, shortest path length, betweenness centrality and clustering coefficient [21, 27]. These measures are presented below, with the scope of the proposed model that was stated above, making the appropriate modifications. It should be noticed again that for the analysis of metabolic pathways and networks, only the main compounds are taken into account, omitting the cofactors from consideration. This is an approach for getting biologically meaningful results and making the topological analysis more realistic. The following measures are all implemented in VisBolic tool.

4.1 Parameters related to Shortest Paths

In a hypergraph the length of a hyperpath, (simple called as *path*), is the number of hyperedges forming it. There may be multiple paths connecting two given hypernodes. The shortest path length between two hypernodes u_{h1} and u_{h2} is defined as their **distance** and is denoted by L(u_{h1},u_{h2}). The maximum distance between any two hypernodes is termed as the **diameter of the network**. If a network is disconnected, its diameter is the maximum of all diameters of its connected components. The **network radius** on the other hand is the minimum distance between any two hypernodes.

The **average shortest path length** or the **characteristic path length** gives the expected distance between two hypernodes and is a measure of network's overall navigability. Finally, the **shortest path length distribution** gives the number of hypernode pairs v_{h1}, v_{h2} with L(v_{h1},v_{h2})=k for k=1,2,... The network diameter and the shortest path length distribution may indicate small-world properties of the analysed network, in which any two hypernodes can be connected with relatively shortest paths. Many real world networks have a small world architecture [79].

4.2 Centrality Measures

4.2.1 Degree Centrality

In directed hypergraphs H = (V,E_h) with V_h the set of hypernodes and E_h the set of hyperedges, the in-degree of a hypernode $v_h \in V_h$, denoted by $deg^+(v_h)$, is defined as below:

$$deg^+(v_h) =\mid \{e_{hi} : e_{hi} \in E_h \bigwedge head(e_{hi}) = v_h\} \mid$$

Similarly, the out-degree of a hypernode $v_h \in V_h$, denoted by $deg^-(v_h)$, is defined as below:

$$deg^{-}(v_h) =\mid \{e_{hi} : e_{hi} \in E_h \bigwedge tail(e_{hi}) = v_h\} \mid$$

The sum of in and out degree of a hypernode v_h is its degree. Self-loops are counted for the computation of hypernode degrees. The **degree sum formula** for directed hypergraphs states that:

$$\sum_{v_h \in V_h} deg^{+}(v_h) = \sum_{v_h \in V_h} deg^{-}(v_h) =\mid E_h \mid$$

A hypernode with high degree, known as *hub*, is better connected to the network and therefore may play a more important role in maintaining the network structure. Comparative analysis of metabolic networks of 43 different organisms, revealed the same set of highly connected metabolites indicating the generic utilization of the same compounds by each organism [43].

The **degree distribution** P(k) of a network is defined to be the fraction of hypernodes in the network with degree k and expresses the probability that a selected hypernode v_h has degree equal to k. The degree distribution is estimated by counting the number of hypernodes with degree k and dividing by the total number of hypernodes. Besides, there are in-degree and out-degree distribution. Degree distribution have been extensively studied in theoretical network models and have been used to distinguish between different classes of network topologies, i.e., between random and scale-free network topologies [15].

4.2.2 Betweenness Centrality

Betweenness centrality is another centrality measure of a hypernode within a network. Hypernodes that occur on many shortest paths between other hypernodes have higher betweenness than those that do not. Betweenness centrality is defined as:

$$C_B(u_h) = \sum_{s \neq u_h \neq t} \frac{\sigma_{st}(u_h)}{\sigma_{st}}$$

where s and t are hypernodes in the network different from u_h, σ_{st} denotes the number of shortest paths from s to t, and $\sigma_{st}(u_h)$ is the number of shortest paths from s to t that u_h lies on.

This measure is normalised by dividing by the number of hypernodes pairs not including u_h, which is $(|V_h| - 1)(|V_h| - 2))$, where $|V_h|$ is the number of hypernodes in the connected component that u_h belongs to.

4.2.3 Closeness Centrality

Closeness centrality is one of the basic concepts in a topological space. Intuitively we say that two sets are close if they are arbitrarily near to each other. Hypernodes that tend to have short distances to other hypernodes have higher closeness centrality. Closeness centrality of a hypernode u_h is defined as the mean shortest distance between the hypernode and all other hypernodes reachable from it:

$$C_C(u_h) = \frac{\sum_{t \in V_h \setminus u_h} L(u_h, t)}{|V_h| - 1}$$

where $|V_h| > 2$ is the size of the connected component u_h belongs to. Closeness can be regarded as a measure of how long it will take information to spread from a given vertex to other reachable vertices in the network. The closeness centrality of hypernodes that belong to isolated hyperedges is 0.

4.3 Parameters related to Neighbourhood

4.3.1 Neighbourhood Connectivity

The neighbourhood of a hypernode v_h is defined as:

$$N_{v_h} = \{u_h : e_{v_h u_h} \in E_h \bigwedge e_{u_h v_h} \in E_h\}$$

The **connectivity** of v_h, denoted by k_{v_h}, is the size of its neighbourhood. Self-loops are ignored. The **neighbourhood connectivity** of a hypernode v_h is defined as the average connectivity of all its neighbours of v_h. The **average number of neighbours** indicates the **average connectivity** of a hypernode in the network. The **neighbourhood connectivity distribution** gives the average of the neighbourhood connectivities of all hypernodes v_h with k neighbours, for k=1,2,...

In analogy to the in and out degree, every hypernode v_h has in and out connectivity. Thus there are the following types of neighbourhood connectivity:

- Only in - The average out-connectivity of all in-neighbours of v_h
- Only out - The average in-connectivity of all out-neighbours of v_h
- In and Out - The average connectivity of all neighbours of v_h

4.3.2 Network Density

A normalized version of the average number of neighbours is **network density**. The maximum number of hyperedges in a hypergraph H=(V, E_h) with V_h the set of hypernodes is $|V_h||V_h - 1|$. A dense hypergraph is a hypergraph with number of hyperedges close to maximal number. For hypergraphs the density D is defined as:

$$\mathrm{D(H)} = \frac{|E_h|+|E_p|}{|V_h||V_h-1|}$$

and is normalized to $[0, 1]$. Self-loops and multiple hyperedges are ignored. A hypergraph is sparse, if $\mathrm{D} \ll 1$. A hypergraph with isolated hyperedges has density of 0.

4.3.3 Shared Neighbours

$P(u_h, v_h)$ is the number of shared partners between two hypernodes u_h and v_h, i.e. hypernodes that are neighbours of both u_h and v_h. The shared neighbours distribution gives the number of hypernode pairs (u_h, v_h) with $P(u_h, v_h) = \mathrm{k}$, for k = 1,2,... Pseudo-edges are ignored.

4.3.4 Topological Coefficients

The topological coefficient T_{u_h} of a hypernode u_h with k neighbours is computed as follows:

$$T_{u_h} = \frac{avg(J(u_h, v_h))}{k}$$

$J(u_h, v_h)$ is defined for all hypernodes v_h that share at least one neighbour with u_h. The value $J(u_h, v_h)$ is the number of neighbours shared between the hypernodes u_h and v_h, plus one if there is a direct link between u_h and v_h.

The topological coefficient is a relative measure for the extent to which a hypernode shares neighbours with other hypernodes. Note that hypernodes with one or no neighbours are assigned a topological coefficient of 0. Pseudo-edges are ignored.

4.4 Clustering Coefficient

The clustering coefficient is a measure of the cliquishness of a neighbourhood and takes value within [0,1] [79, 15]. Two versions of this measure exist: the global and the local. The **local clustering coefficient** of a hypernode v_h quantifies how close its neighbours are to being a clique and is given by the proportion of links between the hypernodes within its neighbourhood (in and out) divided by the number of links that could possibly exist between them, i.e.,

$$C_{v_h} = \frac{|\{e_{u_h k_h}\}|}{|N_{v_h}|(|N_{v_h}|-1)} : u_h, k_h \in N_{v_h}, e_{u_h k_h} \in E_h$$

Hypernodes with less than two neighbours are assumed to have a clustering coefficient of 0.

The **network clustering coefficient** is given as the average of the local clustering coefficients of all hypernodes and expresses the overall tendency of hypernodes to participate in clusters. The **average clustering coefficient distribution** gives the average of the clustering coefficients for all hypernodes with k neighbours for k=2,3,... In [62] the average clustering coefficient distribution is used to identify a modular organization of metabolic networks while a high clustering coefficient is another indicator of a small world.

4.5 Graph Analysis Algorithms

The algorithms for the computation of traversals, components and shortest paths of graphs were developed during the decades of the 1970s to 1980s and are used till now for the analysis of graph properties [21, 38, 73]. These algorithms could be easily modified and implemented for hypergraphs. The main modification regards the consideration of hypernodes instead of simple nodes, and the consideration of pseudo-edges that exist between hypenodes of the hypergraph. Therefore, the complexity of the following algorithms does not differ, when the input is a hypergraph.

Traversals: Graph traversal refers to the problem of systematic visiting all the nodes of a graph in a particular manner. Breadth First Search (BFS) and Depth First Search (DFS) are the two basic algorithms for searching a graph [21]. BFS visits the sibling nodes before visiting the child nodes, while DFS visits the child nodes before visiting the sibling nodes. BFS computes also the distance from a source node to each reachable vertex. BFS and DFS could be easily modified for hypergraphs, regarding hypernodes instead of nodes and taking into account the pseudo-edges that exist in the hypergraph.

Topological ordering: A topological sorting or a topological ordering of a graph G = (V, E) is a linear ordering of all its nodes such that if G contains an edge (u,v), then u appears before v in the ordering [21]. If graph contains cycles, then no linear ordering is possible. A topological sort of a graph can be viewed as an ordering of its nodes along a horizontal line so that all directed edges go from left to right. DFS can be used to perform a topological sort of a graph or hypergraph.

Cycle Detection: A *hyperpath* is defined as a sequence $P = \{u_1, e_1, u_2, e_2, ..., e_{n-1}, u_n\}$, where $\forall$ i, $1 \leq i \leq n-1 : u_i$ is a hypernode, e_i is a hyperedge, $u_i = Tail(e_i)$ and $u_{i+1} = Head(e_i)$. A hyperpath is a hypercycle if $u_n \bigcap u_1 \neq \emptyset$. Therefore, in order to detect a hypercycle in a hypergraph, it is necessary to check for non-empty intersection, in contrast with the simple equality checking required for detecting cycles in directed graphs. For example consider the hyperpath P = $\{\{G\}, e_5, \{E, F\}, e_3, \{C, B\}, e_2, \{A\}, e_4, \{F\}\}$ in Figure 4.1. Since $\{E, F\} \bigcap \{F\} \neq \emptyset$, the hyperpath P is a hypercylcle. DFS can be used for the detection of cycles in hypergraphs: every time the algorithm detects a back edge, a basic cycle has been revealed.

All-pairs shortest paths: The all-pairs shortest paths refers to the problem of determination of the shortest graph distances between every pair of nodes in a given graph. The problem can be solved using various algorithms, the most common is Floyd-Warshall algorithm [21], which is an example of dynamic programming. Floyd-Warshall algorithm could be easily modified for hypergraphs, regarding hypernodes instead of nodes and taking into account the pseudo-edges that exist in the hypergraph.

Connected Components: In a graph all nodes that are pairwise connected form a connected component. The number of connected components indicates the connectivity of a network, since a lower number of connected components suggests a stronger connectivity. A graph is called

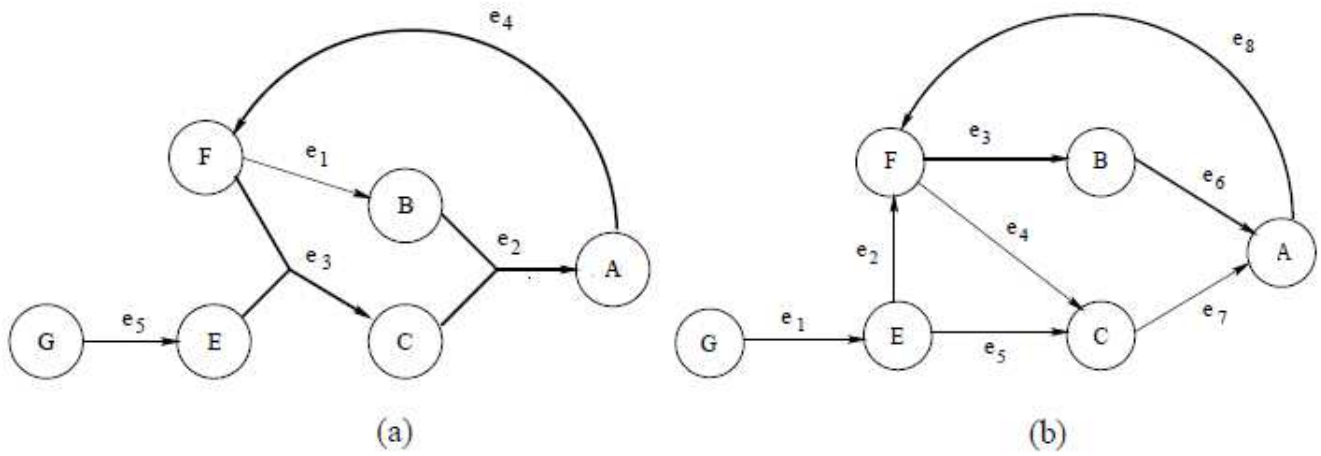


Figure 4.1: *(a) Hypergraph with hypercycles (marked in bold) (b) Directed graph with cycles (marked in bold)*

connected, if it has one connected component, consisting of the whole graph. Connected components of a graph or hypergraph can be computed in linear time using BFS or DFS algorithm. A search that begins at some particular hypernode u_h will find the entire connected component containing u_h before returning. To find all the connected components of a hypergraph, loop through its hypernodes, starting a new breadth first or depth first search whenever the loop reaches a hypernode that has not already been included in a previously found connected component.

Strongly Connected Components: A strongly connected component of a graph is a maximal set of nodes $C \subseteq V$ such that for every pair of nodes u and v there is a directed path from u to v and a directed path from v to u. If each strongly connected component is contracted to a single graph, the resulting graph is acyclic. Therefore, a directed graph is acyclic if and only if it has no non-trivial strongly connected sub-graphs (because a cycle is strongly connected, and every strongly connected graph contains at least one cycle). A graph is strongly connected if it has one strongly connected component, consisting of the whole graph. The most common

algorithm for the computation of the strongly connected components of a graph is the Tarjan's algorithm [73]. This algorithm can be easily modified for the computation of strongly connected components in hypergraphs, regarding hypernodes instead of nodes and taking into account the pseudo-edges that exist in the hypergraph.

Biconnected Components: An articulation point or a cut vertex is a node whose removal disconnects the graph. A graph is biconnected if it has no articulation points. In other words, a biconnected graph is connected and non-separable, meaning that if any node were to be removed, the graph will remain connected. This property is especially useful in maintaining a graph with a two-fold redundancy, to prevent disconnection upon the removal of a single edge. The most common algorithm for the computation of the biconnected components of a graph is the Tarjan's algorithm [73]. This algorithm can be easily modified for the computation of biconnected components in hypergraphs, regarding hypernodes instead of nodes and taking into account the pseudo-edges that exist in the hypergraph.

Chapter 5

Visualization of Metabolic Pathways

A large amount of data on metabolic pathways is available in various databases. A graphical representation of such networks is useful for managing the intrinsic complexity of the data. While the idea is a simple one, the actual creation of complex pathway drawings is not a simple task. Metabolic pathways are typically modelled as graphs, in which nodes represent metabolites while edges represent chemical reactions between metabolites. Usually the enzymes catalysing the reaction are displayed as edge labels. Thus, the problem of visualizing pathways can be formulated as a graph layout problem. Some existing approaches use fixed, pre-computed drawings of metabolic networks and map the data into predefined positions.
Such an example is KEGG, which visualizes pathways in a static way [45]. Pathway diagrams are manually drawn and stored as image maps with links to additional information, on enzymes and adjacent pathways. Another example of static visualization is the ExPASy Molecular Biology Server [36], which gives on-line access to the scanned version of the Boehringer Mannheim Biochemical Pathways map [58]. The map is partitioned in 115 rectangular pieces. Keywords entered by the user are matched against entries on the map, and the corresponding pieces of the map can be displayed.

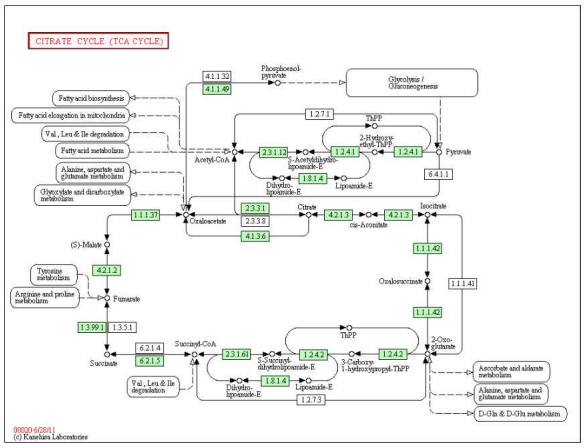


Figure 5.1: *KEGG example*

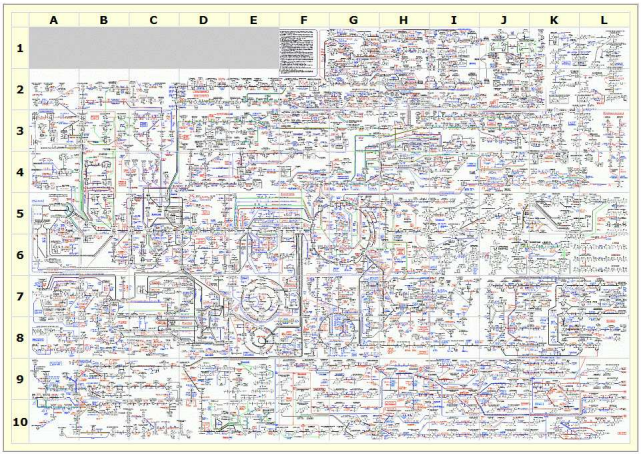


Figure 5.2: *ExPASy*

Static visualization of pathways has several drawbacks. Whenever the data have been updated, the corresponding image has to be edited manually to reflect the changes. Also, there is no way to specify the amount of detail to be displayed. When it has to visualize novel pathways, static visualization is not applicable at all. Therefore the visualization should be performed dynamically at runtime, based on the information provided.

Various algorithms have been used for the visualization of metabolic pathways and networks ranging from the general and standard to special-purpose graph drawing algorithms. The standard graph drawing algorithms include algorithms for circular, hierarchical, orthogonal or force directed layout [25]. However none of these algorithms give satisfactory results when applied to networks without any modification and are not sufficient to fulfil the special requirements and graphical conventions for metabolic networks. On the other side, the special-purpose pathway drawing algorithms are usually applicable to pathways or networks with specific topology. A review of these algorithms is described briefly below.

5.1 Related Work

Little previous work has been done on developing special graph layout algorithms for drawing biochemical networks. Karp and Paley in [47] have pointed out that rather than searching for a single, all-purpose layout algorithm, different algorithms should be applied to parts of the network with different topologies. They devised an algorithm which breaks the graph into cyclic, linear and tree-structured components and then applies different layout methods to each of them individually. Each component is treated as a super-node that is positioned recursively by another layout algorithm. Once positions for main nodes have been determined, another algorithm computes positions for cofactors and enzymes. This algorithm has been implemented in the EcoCyc system [49].

Similarly, an algorithm is presented in [17], based on the work of Karp and Paley. The algorithm uses a combination of circular, hierarchic and a special force-directed graph layout algorithm to compute positions of the graph elements representing main metabolites and reactions. The algorithm is particularly designed for cyclic or partially cyclic pathways or for combinations of complex pathways. It has been tested on five different -relatively complex- pathways or combinations of pathways and good results were produced.

The algorithm proposed by Karp and Paley in [47] does not only compute positions for main compounds but also deals with the problem of placing cofactors and enzymes. This last point has been left out in the algorithm in [17], which is concentrated on the placement of the main compounds only.

A method for the visualization of metabolic pathways using three dimensions is proposed in [65] while a method for visualizing a set of related pathways using two and half dimensions is presented in [19]. Interdependent, two-dimensional layouts of each pathway are stacked on top of each other so that biologists get a full picture of subtle and significant differences among

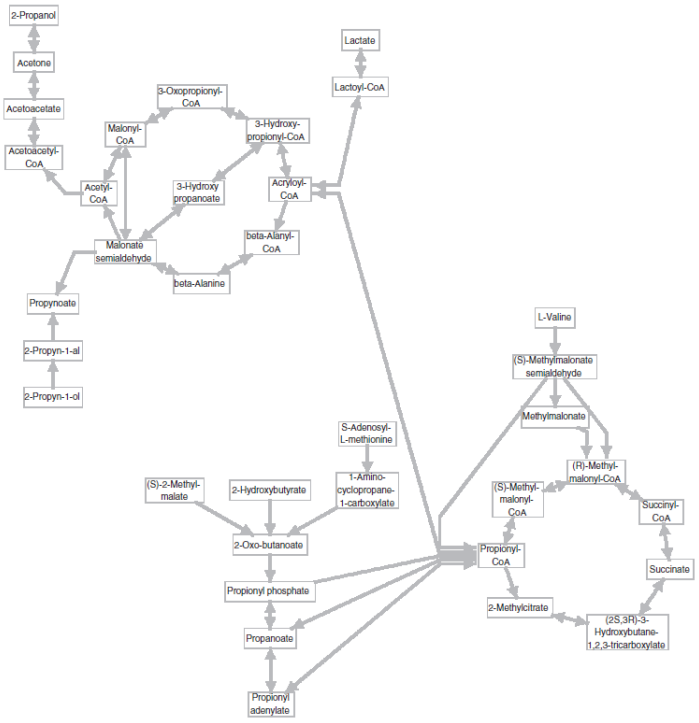


Figure 5.3: *Visualization of Propanoate metabolism using Becker's and Rojas's algorithm*

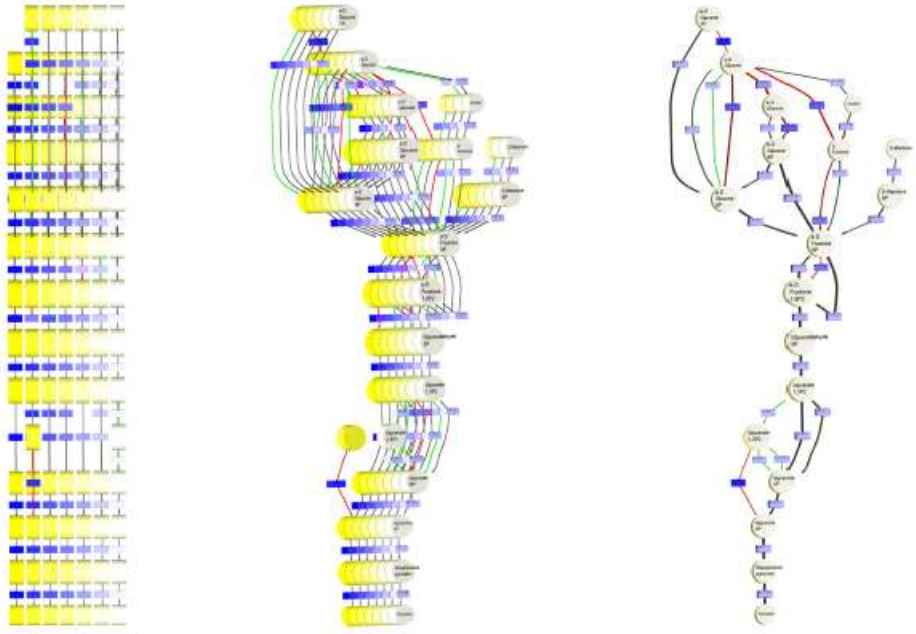


Figure 5.4: *2 1/2D drawing of seven related pathways (parallel projection)*

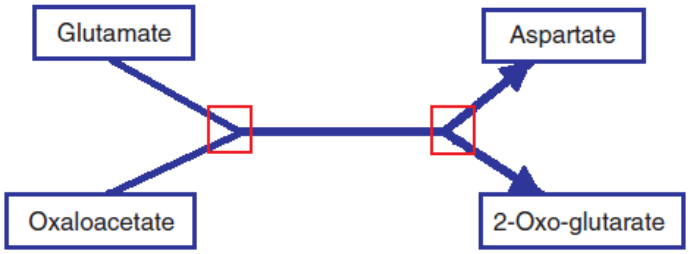


Figure 5.5: *An example of a reaction - Zero size dummy nodes are inserted at the forking positions.*

the pathways. Therefore, the algorithm is performed for the comparison of related pathways.

The above algorithms treat metabolic pathways as graph many functional relationships are lost.

5.2 Proposed Method

The hyperedges could be simulated by inserting a *dummy node* towards the end of the edge which is connected to all source vertices and *another dummy node* at the front of an edge which forks out to the multiple target vertices, as shown in Figure 5.5. It should be noticed that the dummy vertices are inserted at the forking positions, when the hyperedge has multiple source or target nodes. Therefore, two dummy vertices are inserted at most for each hyperedge. If a hyperedge has one source and one target node, it is treated as a common edge. The size of a dummy node is set to zero.

With this approach we force the production of hypergraph-like drawings. In fact these drawings represent graphs that look like hypergraphs. As a result the approach has two basic advantages: we could exploit the variety of existing graph drawing techniques and algorithms for representing metabolic pathways and we could produce hypergraph-like drawings.

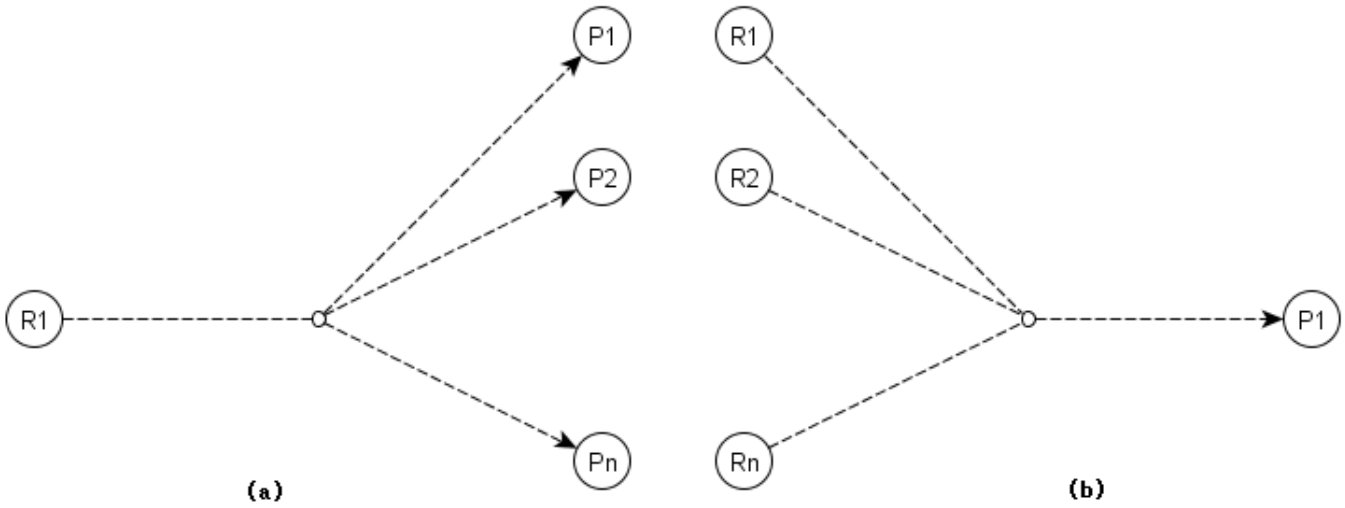


Figure 5.6: *Cases of inserting one dummy vertex: (a) The reaction has one reactant and multiple products; (b) The reaction has multiple reactants and one product. The small white circles represent the dummy vertices.*

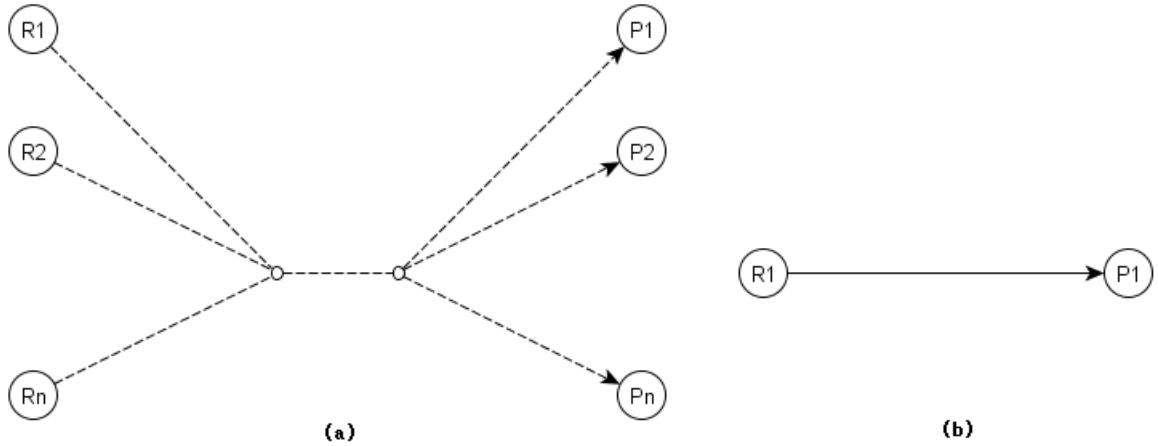


Figure 5.7: *(a) Case of inserting two dummy vertices. The reaction has multiple reactants and multiple products. The small white circles represent the dummy vertices; (b) Case of inserting no dummy vertex.*

5.2.1 Treatment of cofactors in visualization

Cofactors, that are usually highly connected, are allowed to appear in multiple instances of graph nodes. Therefore many "clones" of the same cofactor may occur in a diagram, every time this cofactor exists in a different reaction of the pathway. In fact, this is what is usually done in conventional manual drawings. Otherwise, many edge crossings will occur and the drawing will not be easily readable. For the visualization of a network, this approach is followed. In VisBolic, the multiple occurrences of a cofactor are depicted by grey fill color of the corresponding nodes.

5.3 Results

Visualization of various metabolic pathways using the proposed method, can be seen in following figures. Drawings of these pathways generated by other notable tools such as VANTED, CellDesigner and Cytoscape, are also presented. It should be noticed that the above tools use the bipartite model for representing pathways, while they do not clone cofactors, except VANTED that represent only main compounds.

Based on the following examples, the proposed method for the visualization of metabolic pathways produces a more clear and readable drawing with fewer edge crossings. The use of bipartite model by the other tools has several impacts at the readability of the drawings. Visualization algorithms do not distinguish between two set of nodes, producing more complicated drawings. Also, the readability of the drawings is reduced, since they do not clone cofactors, that participate in different reactions of the pathway, and therefore additional edge crossings are occurred. It should be noticed that the differentiation of our method is based on using dummy vertices and cloned cofactors. The visualization algorithms and the metabolic information are the same for every visualization tool in each case.

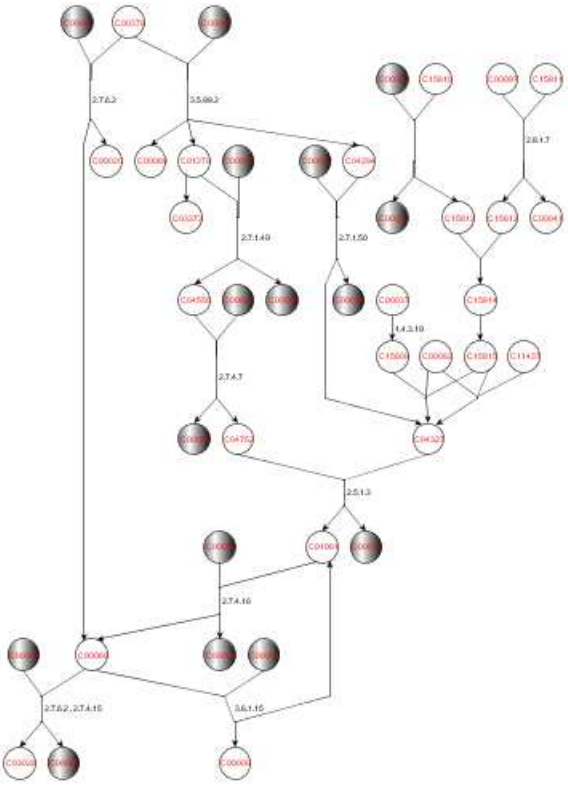


Figure 5.8: *Drawing of Thiamine metabolism of Bacillus subtilis with hierarchical layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 17 reactions and 29 compounds. The drawing contains one edge crossing.*

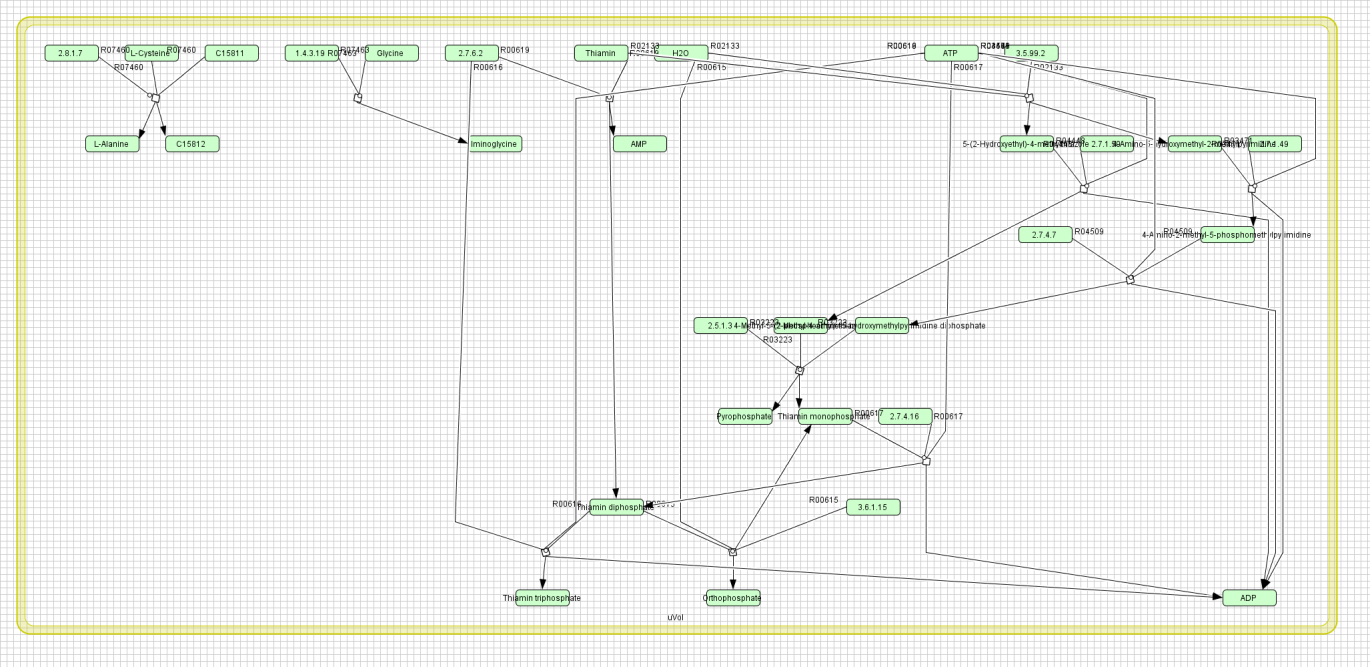


Figure 5.9: *Drawing of Thiamine metabolism of Bacillus subtilis with hierarchical layout algorithm using* ***CellDesigner****. The circles represent reactions whereas the green boxes depict the metabolites and enzymes that catalyse the corresponding reactions.*

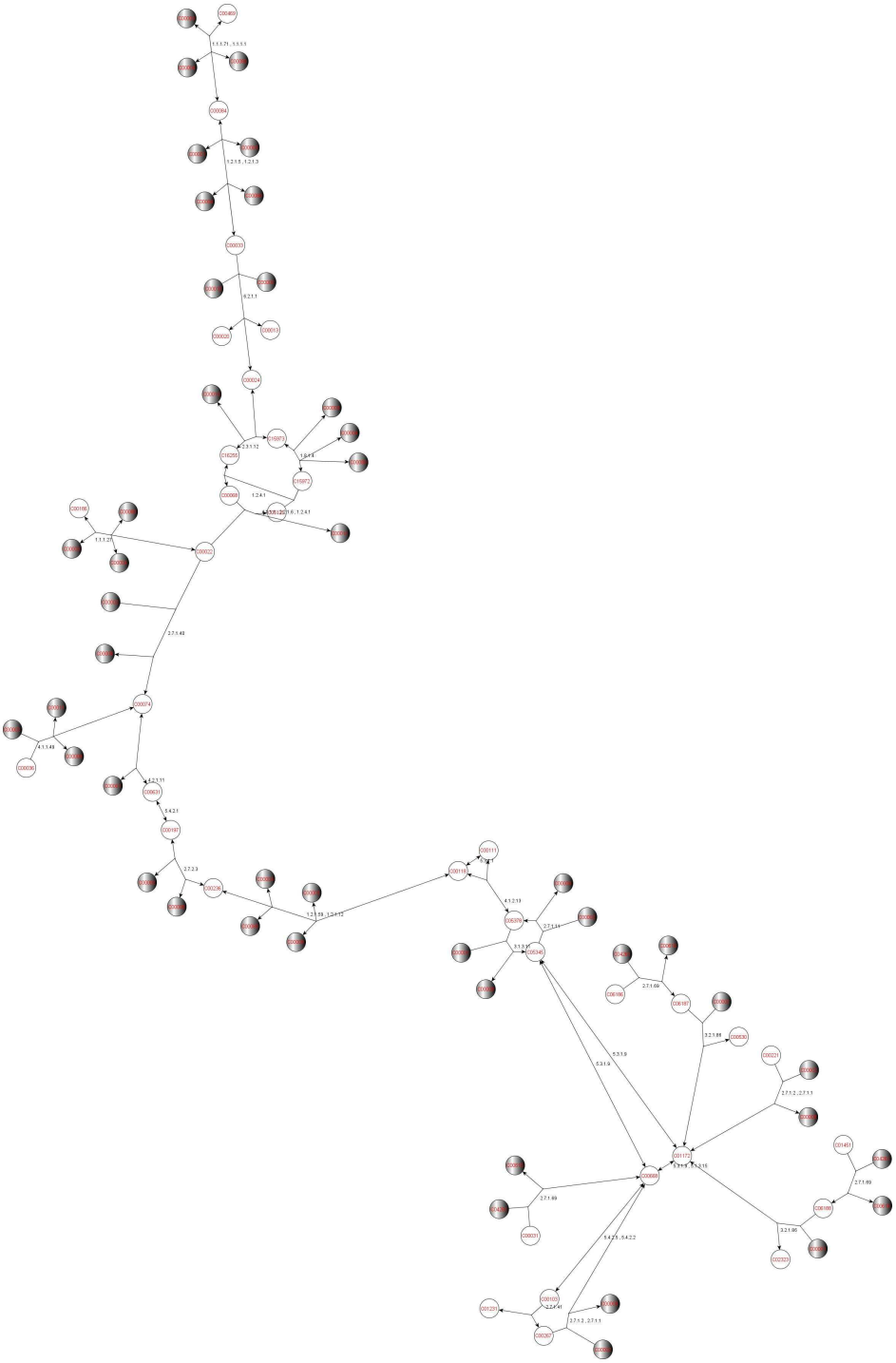


Figure 5.10: *Drawing of Glycolysis / Gluconeogenesis of Bacillus subtilis with circular layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 30 reactions and 46 compounds. The drawing is planar.*

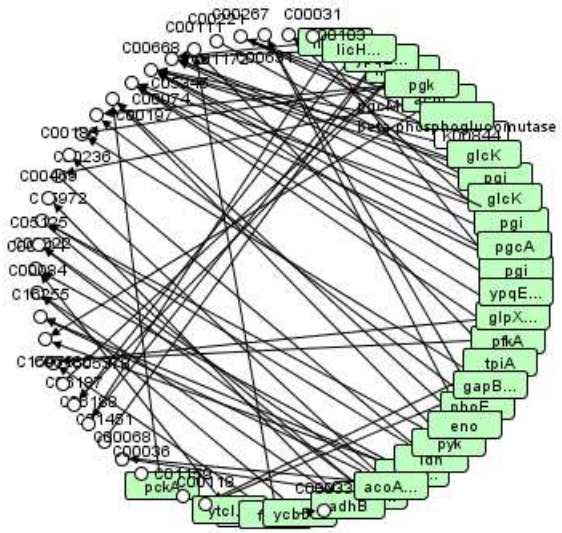


Figure 5.11: *Drawing of Glycolysis / Gluconeogenesis of Bacillus subtilis with circular layout algorithm using* ***VANTED****. The circles represent metabolites whereas the green boxes depict the enzymes that catalyse the corresponding reactions.*

For example, the visualization of the Thiamine metabolism of Bacillus subtilis using VisBolic, produces a drawing with one edge crossing (see Figure 5.8). The visualization of the same pathway, with the same drawing algorithm (hierarchical layout), using CellDesigner produces a less readable drawing with more edge crossings (see Figure 5.9). There are cases where the produced drawing is completely unreadable. For example the drawing of Glycolysis / Gluconeogenesis of Bacillus subtilis with circular layout algorithm using VANTED (see Figure 5.11) or the drawing of Pyruvate metabolism of Saccharomyces cerevisiae with organic layout algorithm using CellDesigner (see Figure 5.16). The proposed method usually produces drawings that are planar or have few edge crossings, where the reactions are clearly visible and separated (see Figure 5.10, 5.12, 5.14, 5.17). However, there are cases that the proposed method does not produce clear drawings. As the size (and the density) of a pathway is getting larger, the less readable the drawing becomes. For example in Figures 5.20 and 5.26, the drawing is almost

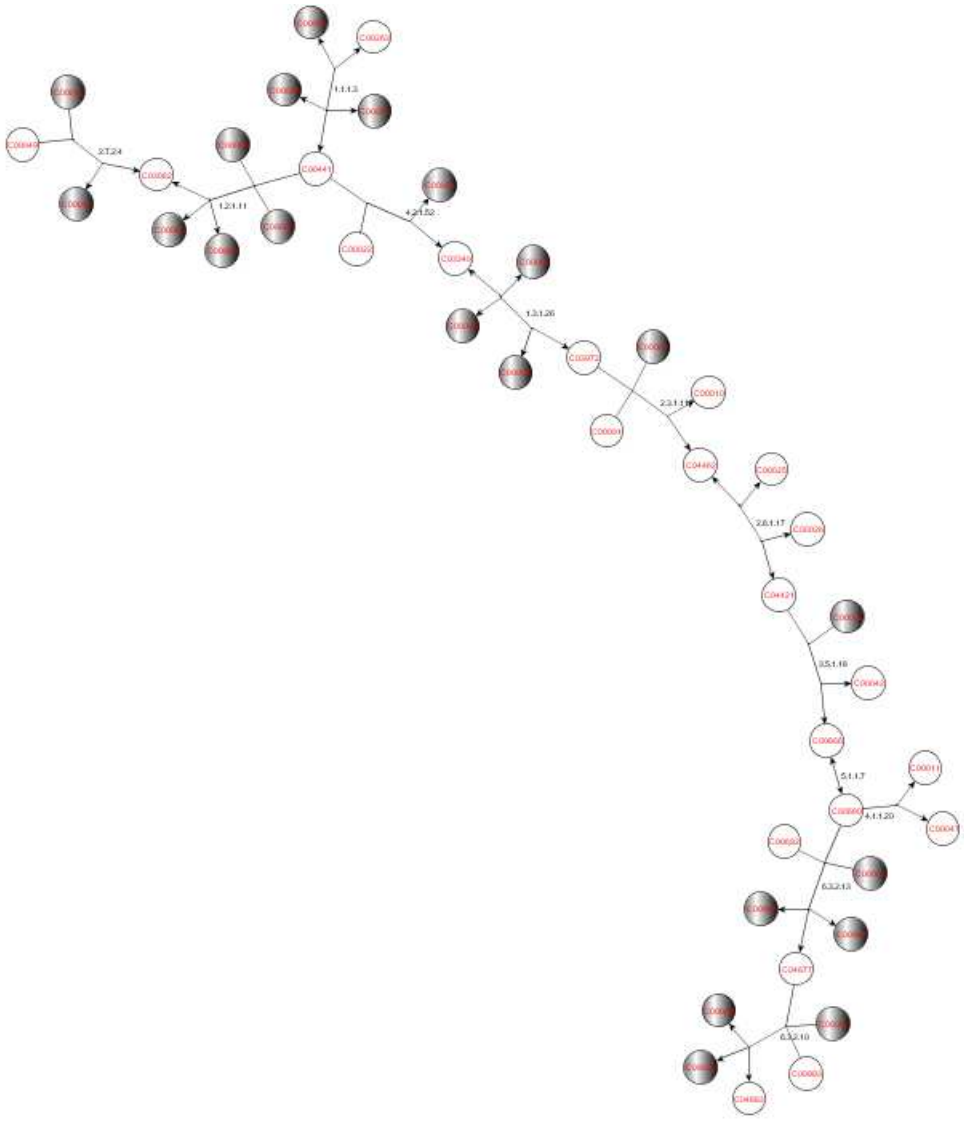


Figure 5.12: *Drawing of Lysine biosynthesis of Escherichia coli K-12 DH10B with organic (smart) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 12 reactions and 29 compounds. The drawing is planar.*

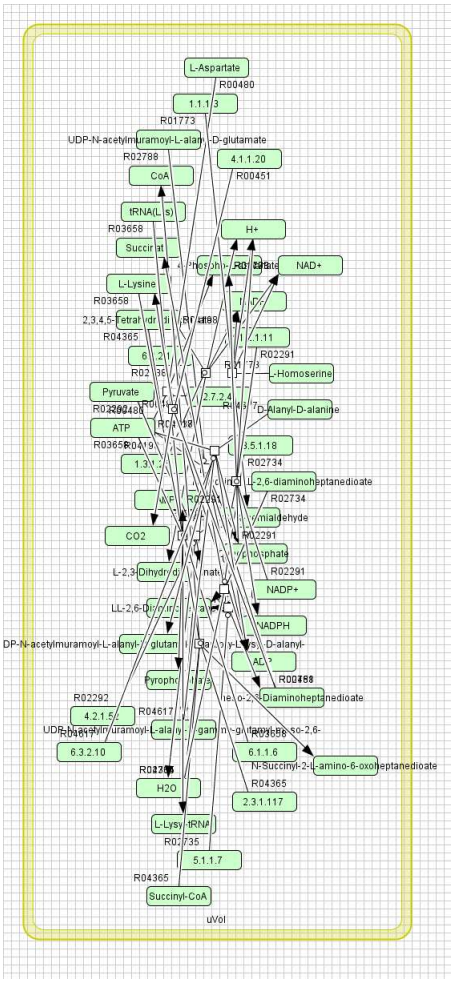


Figure 5.13: *Drawing of Lysine biosynthesis of Escherichia coli K-12 DH10B with organic (smart) layout algorithm using* ***CellDesigner****. The circles represent reactions whereas the green boxes depict the metabolites and enzymes that catalyse the corresponding reactions.*

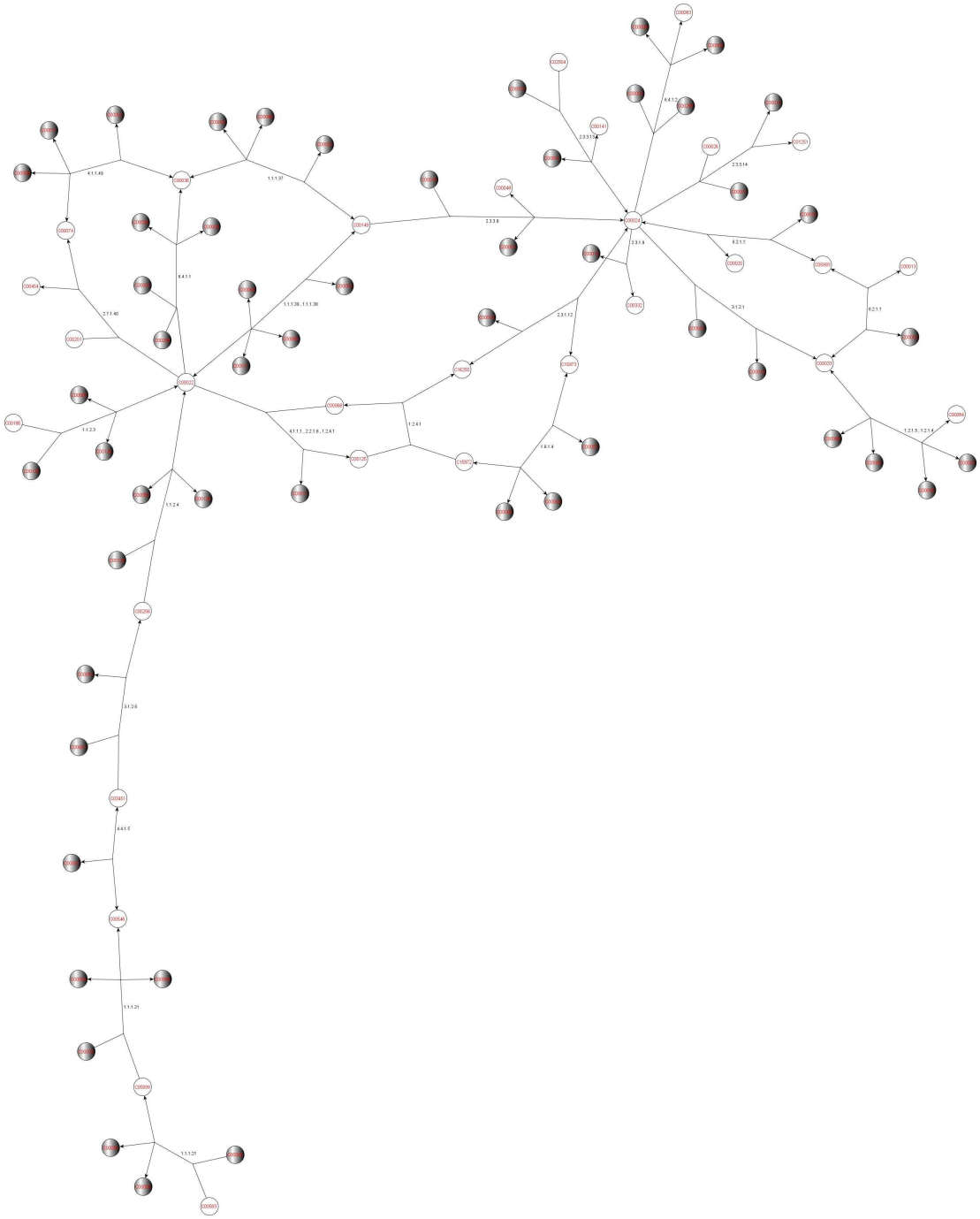


Figure 5.14: *Drawing of Pyruvate metabolism of Saccharomyces cerevisiae (budding yeast) with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 24 reactions and 45 compounds. The drawing is planar.*

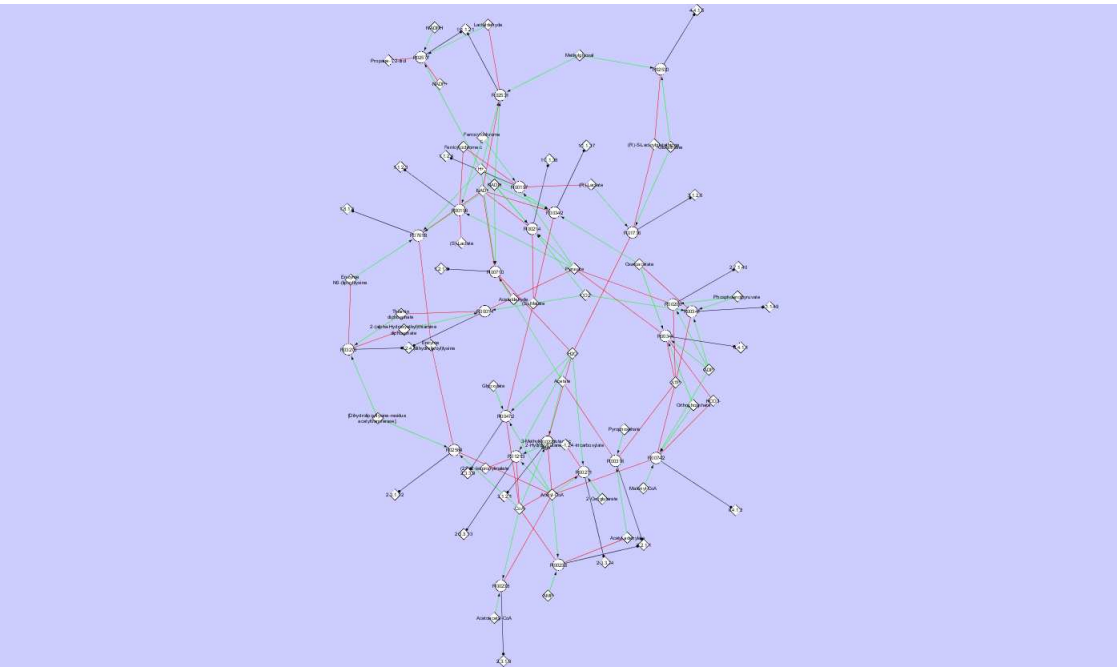


Figure 5.15: *Drawing of Pyruvate metabolism of Saccharomyces cerevisiae (budding yeast) with organic (classic) layout algorithm using* ***Cytoscape****. The circles represent reactions whereas the rhombuses depict the metabolites and enzymes that catalyse the corresponding reactions.*

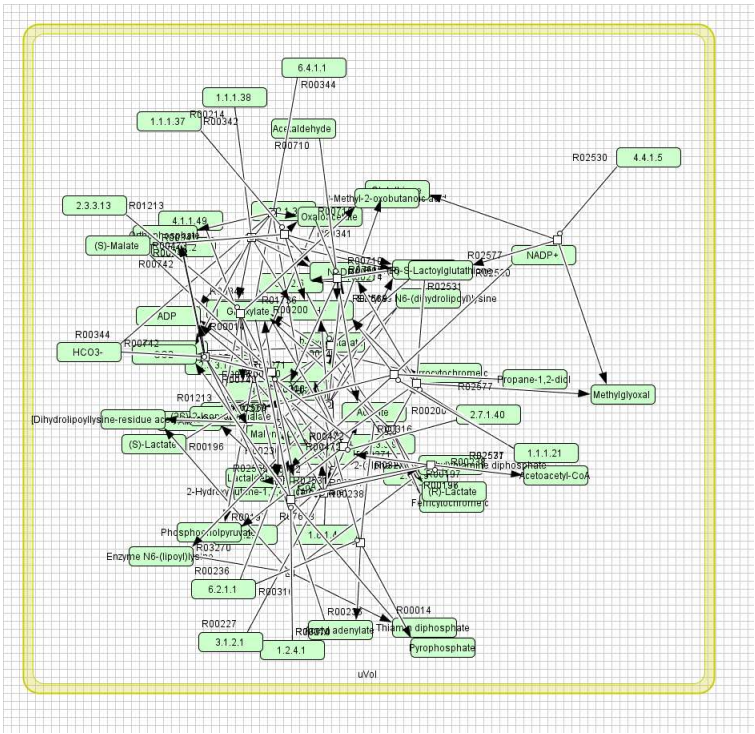


Figure 5.16: *Drawing of Pyruvate metabolism of Saccharomyces cerevisiae (budding yeast) with organic (classic) layout algorithm using* ***CellDesigner****. The circles represent reactions whereas the green boxes depict the metabolites and enzymes that catalyse the corresponding reactions.*

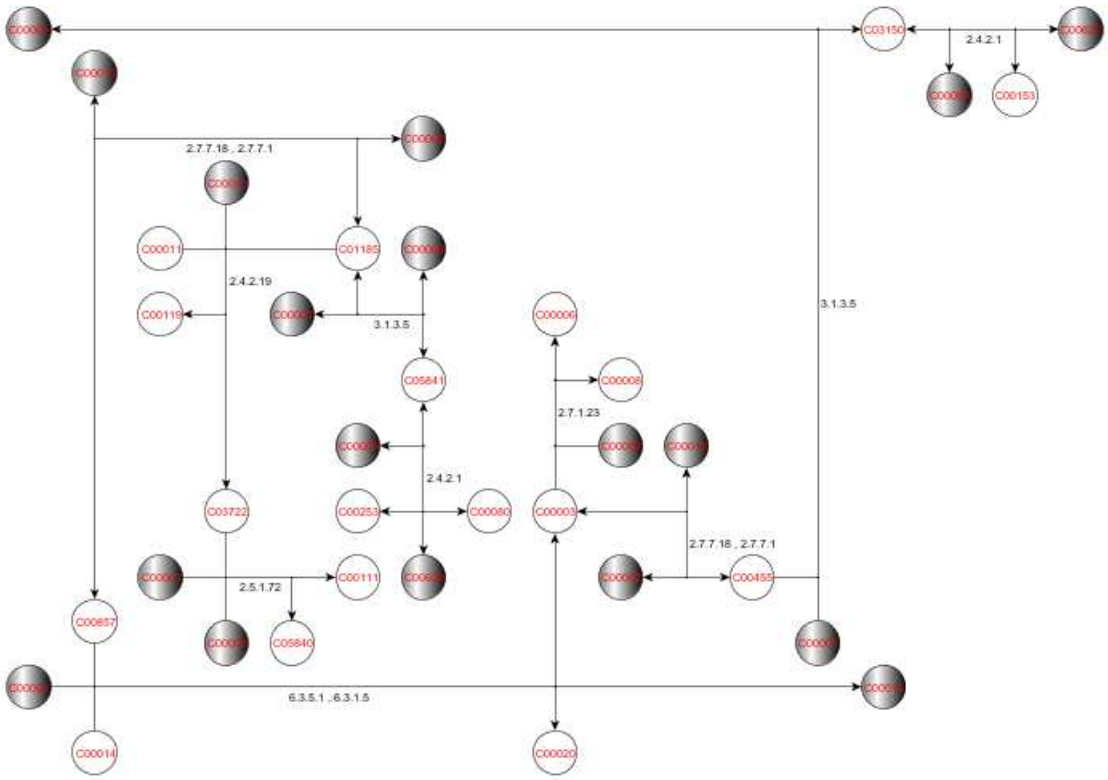


Figure 5.17: *Drawing of Nicotinate and nicotinamide metabolism of Helicobacter pylori P12 with orthogonal (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 10 reactions and 23 compounds.*

unreadable due to the many edge crossings.

It should be noticed that the most appropriate algorithms for visualizing this type of biological networks is proven to be the hierarchical, the organic and the smart organic layout algorithms. The use of classical visualization algorithms has the advantage that specific criteria are optimized, i.e. number of line crossings, number of bends, occupied area, symmetry and length of edges. However, the proposed method does not respect the special drawing conventions that can be found in relevant biochemistry books, e.g. the main direction of pathway is accentuated, reactants and products are closely separated, cofactors are placed out of the main path, close to the reaction while main compounds lie along the backbone of the pathway etc. These domain specific criteria are very important, since biologists are used to read a diagram with a specific way. It should be noticed that the above metrics could be used for a comparative evaluation between the proposed and other existing methods.

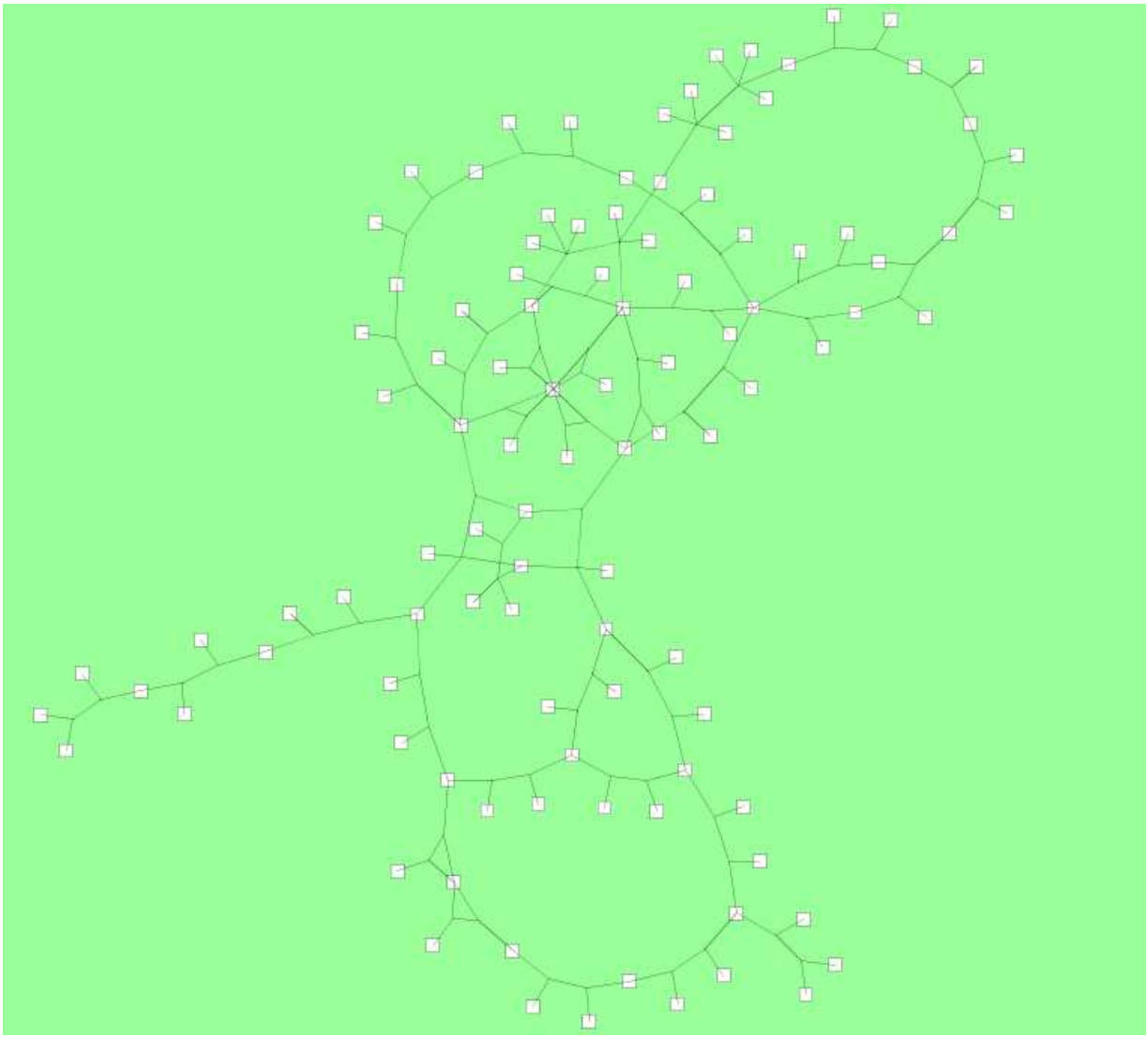


Figure 5.18: *Drawing of Pyrimidine metabolism of Caulobacter crescentus CB15 with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 38 reactions and 49 compounds.*

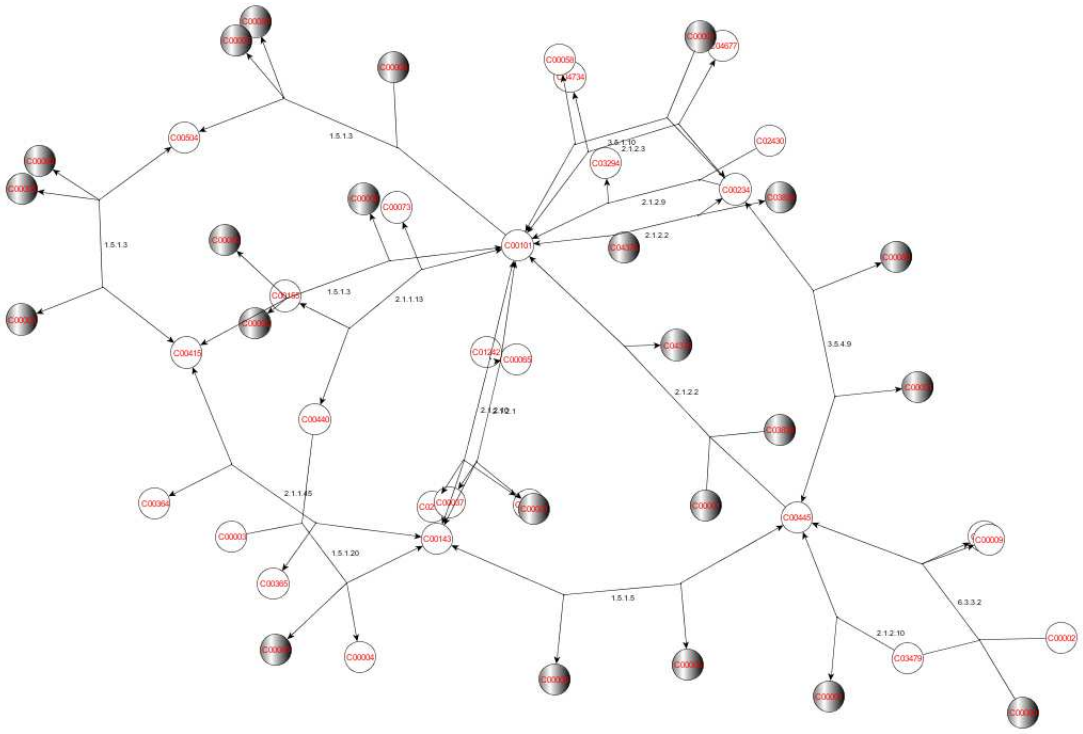


Figure 5.19: *Drawing of One carbon pool by folate of Caulobacter crescentus CB15 with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 17 reactions and 33 compounds.*

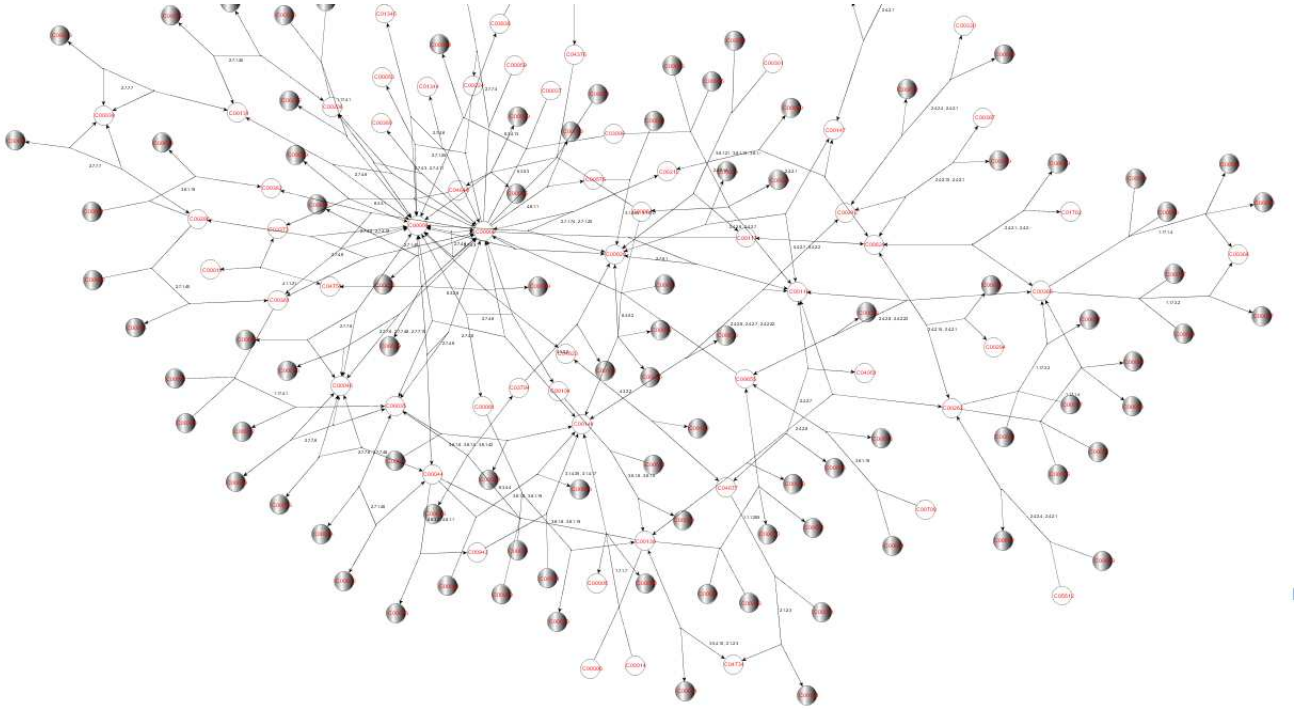


Figure 5.20: *A part of the drawing of Purine metabolism of Caenorhabditis elegans (nematode) with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 70 reactions and 78 compounds.*

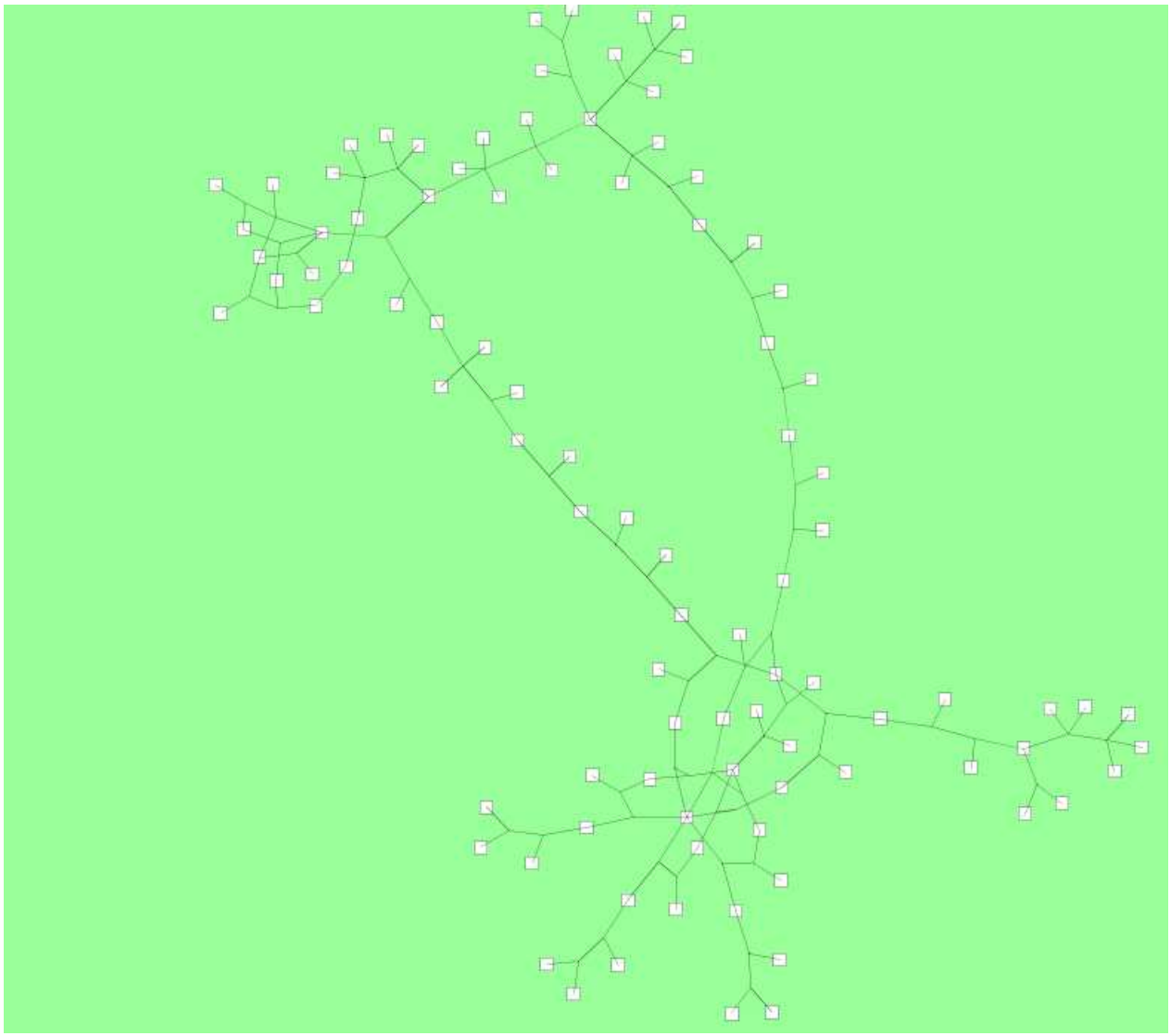


Figure 5.21: *Drawing of Valine, leucine and isoleucine degradation of Caenorhabditis elegans (nematode) with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 34 reactions and 55 compounds.*

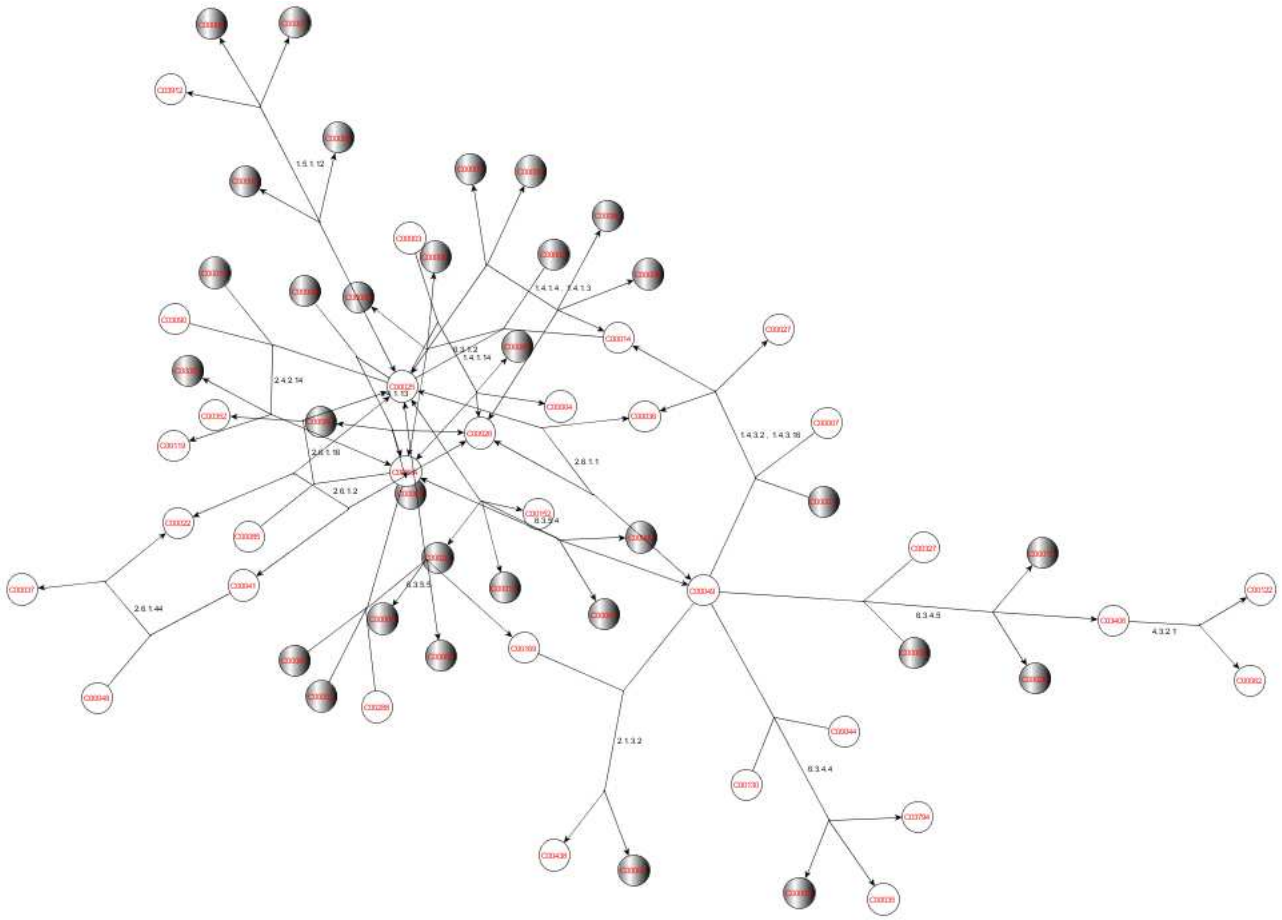


Figure 5.22: *Drawing of Alanine, aspartate and glutamate metabolism of Chlamydomonas reinhardtii with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 17 reactions and 40 compounds.*

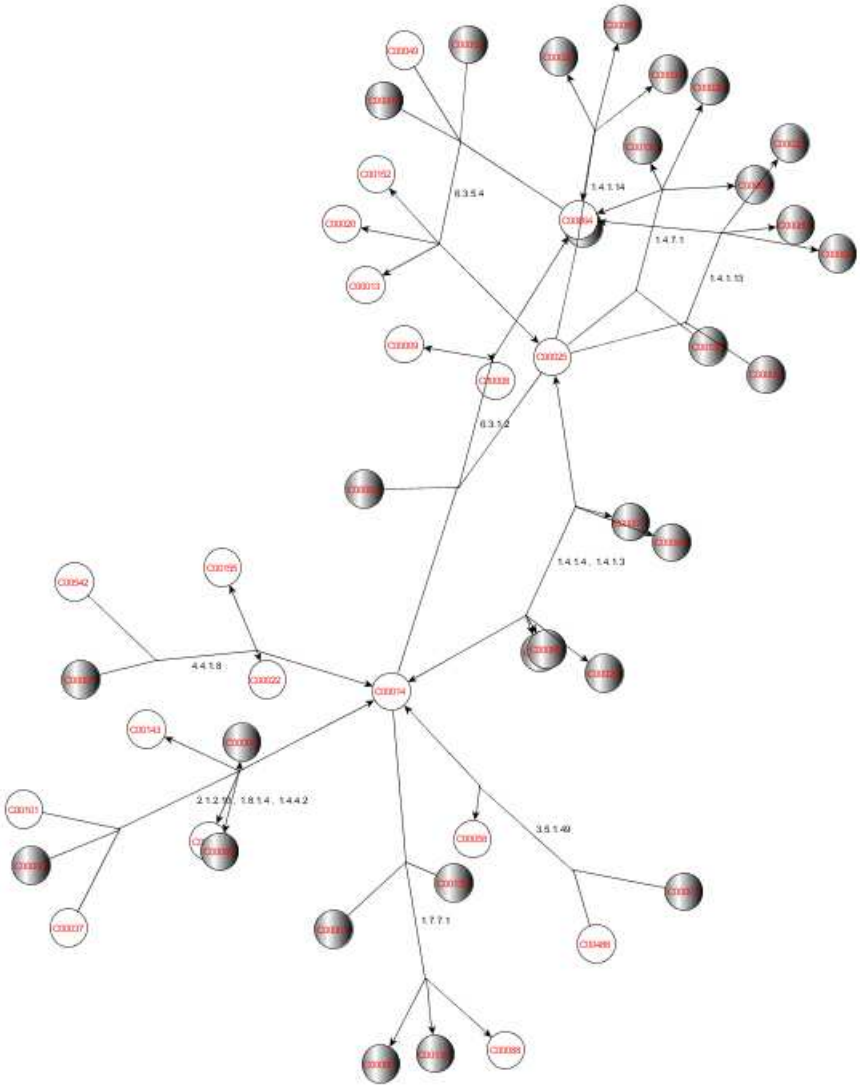


Figure 5.23: *Drawing of Nitrogen metabolism of Chlamydomonas reinhardtii with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 10 reactions and 29 compounds.*

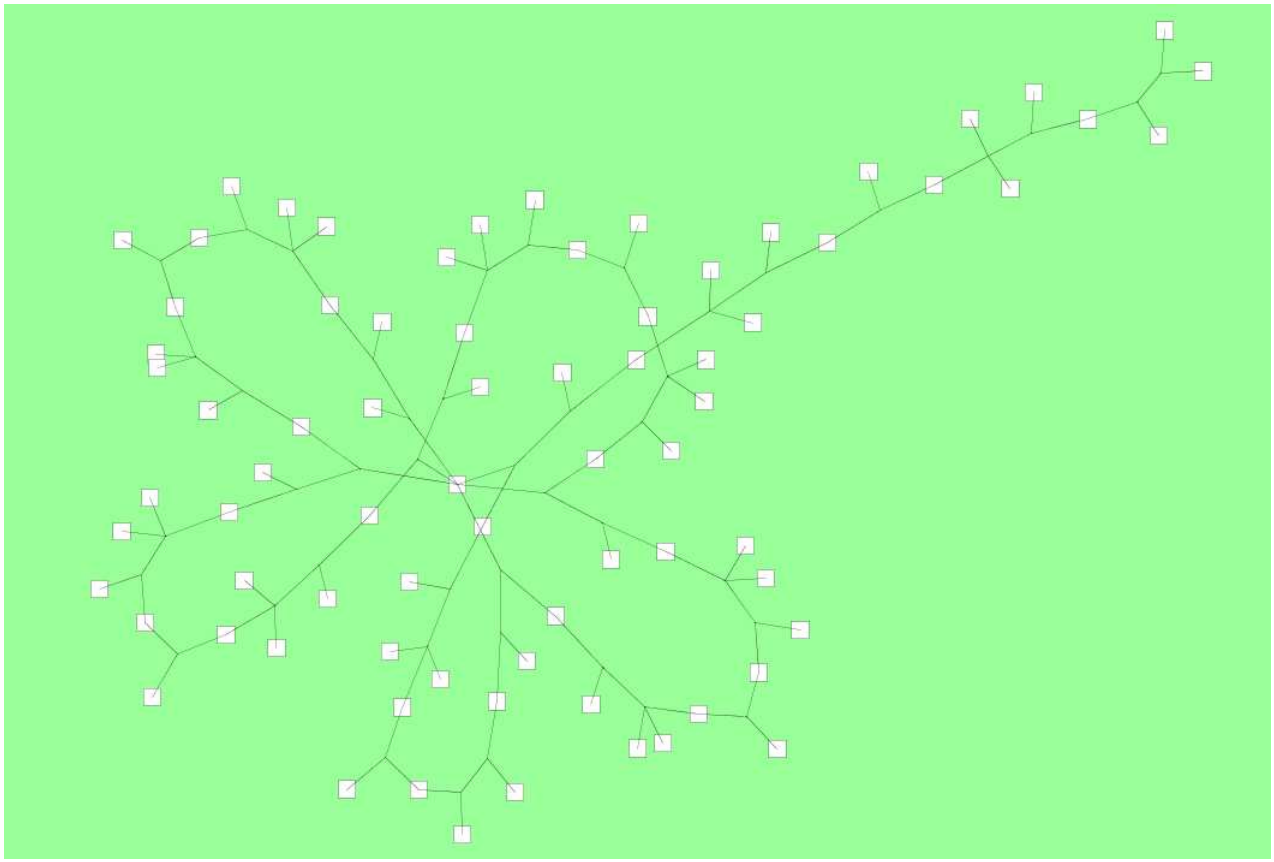


Figure 5.24: *Drawing of Fatty acid elongation in mitochondria of Drosophila melanogaster (fruit fly) with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 25 reactions and 34 compounds.*

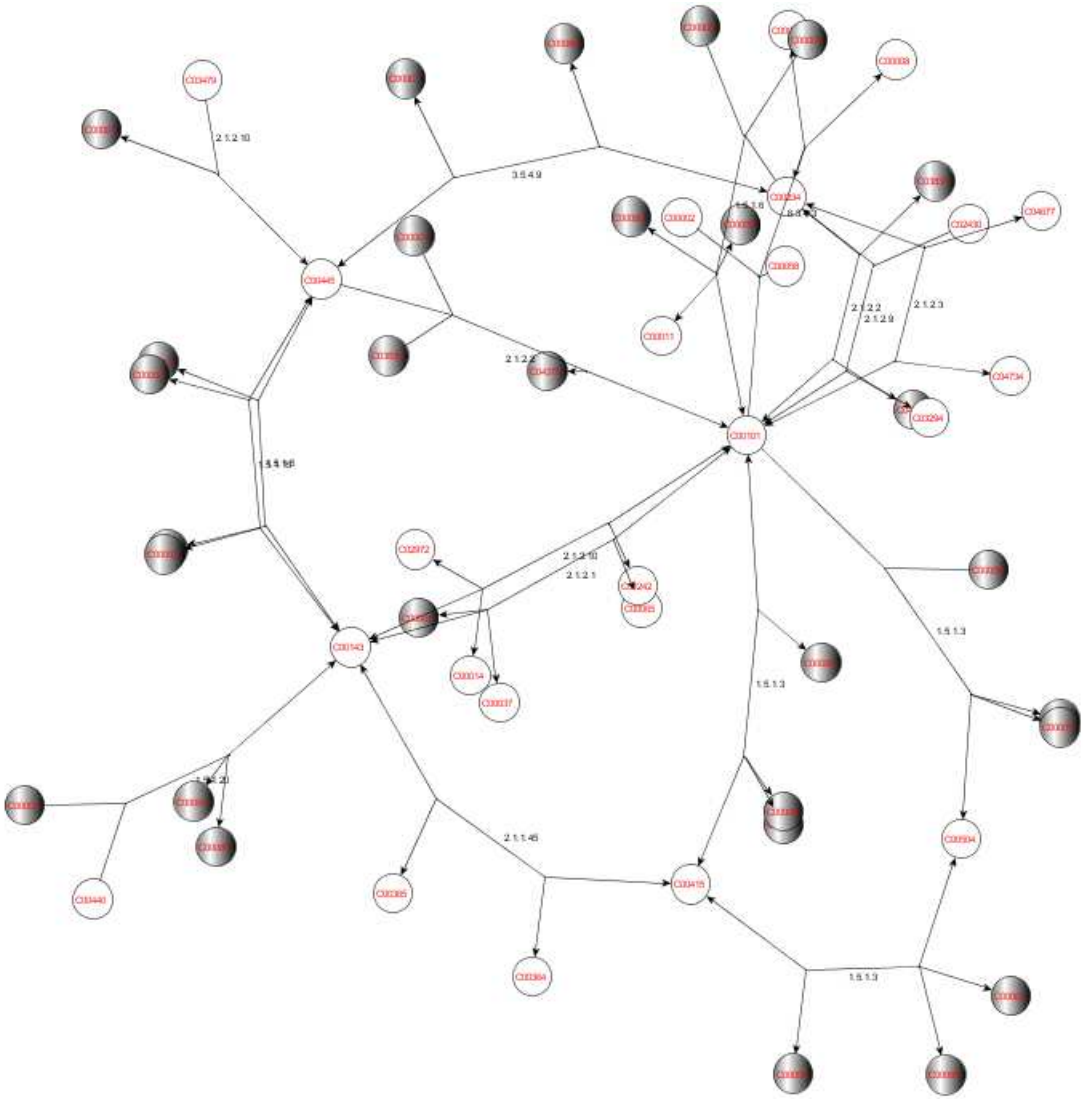


Figure 5.25: *Drawing of One carbon pool by folate of Drosophila melanogaster (fruit fly) with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 17 reactions and 32 compounds.*

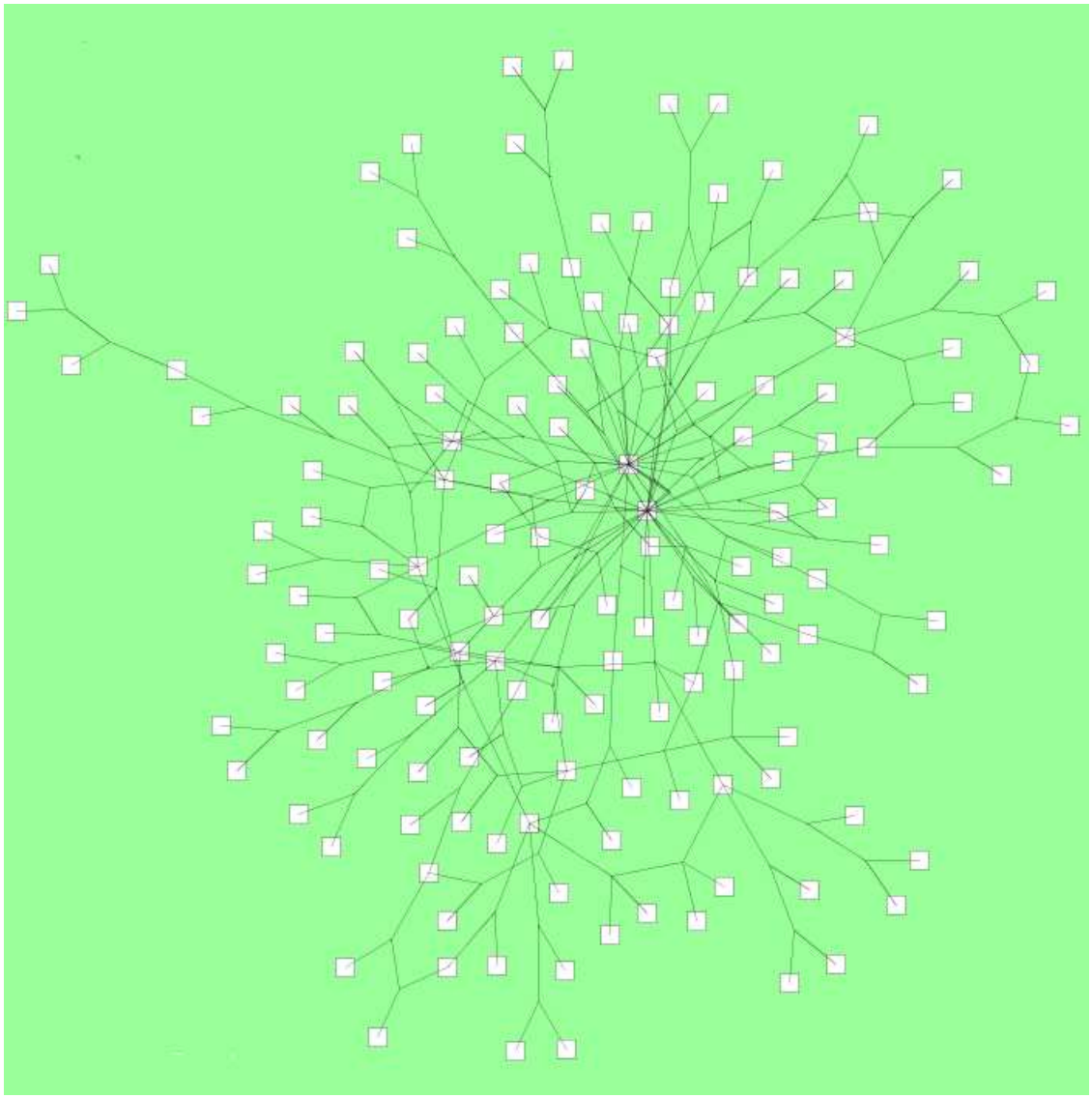


Figure 5.26: *Drawing of Purine metabolism of Synechocystis sp. PCC6803 with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 66 reactions and 77 compounds.*

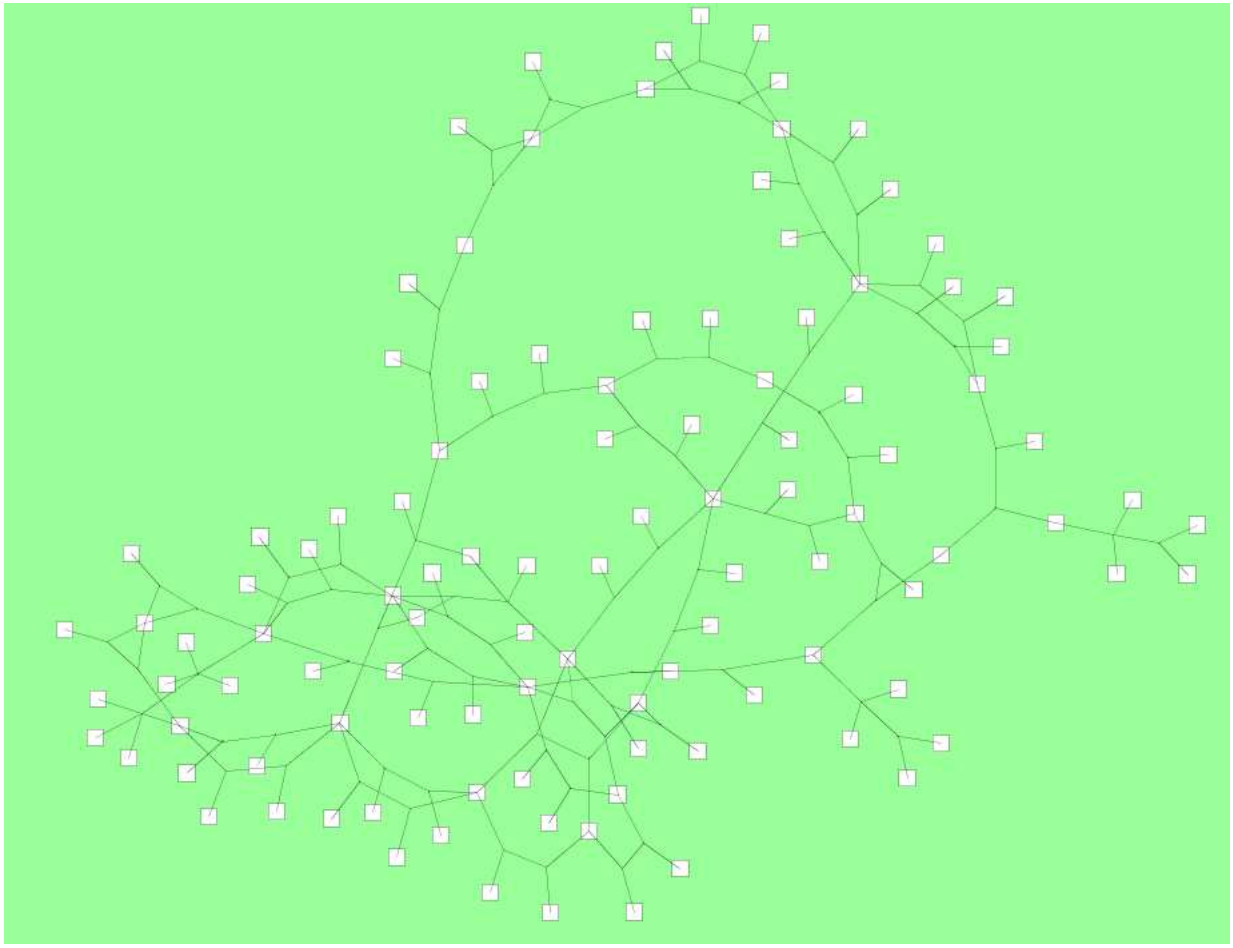


Figure 5.27: *Drawing of Pyrimidine metabolism of Tetrahymena thermophila with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 42 reactions and 43 compounds.*

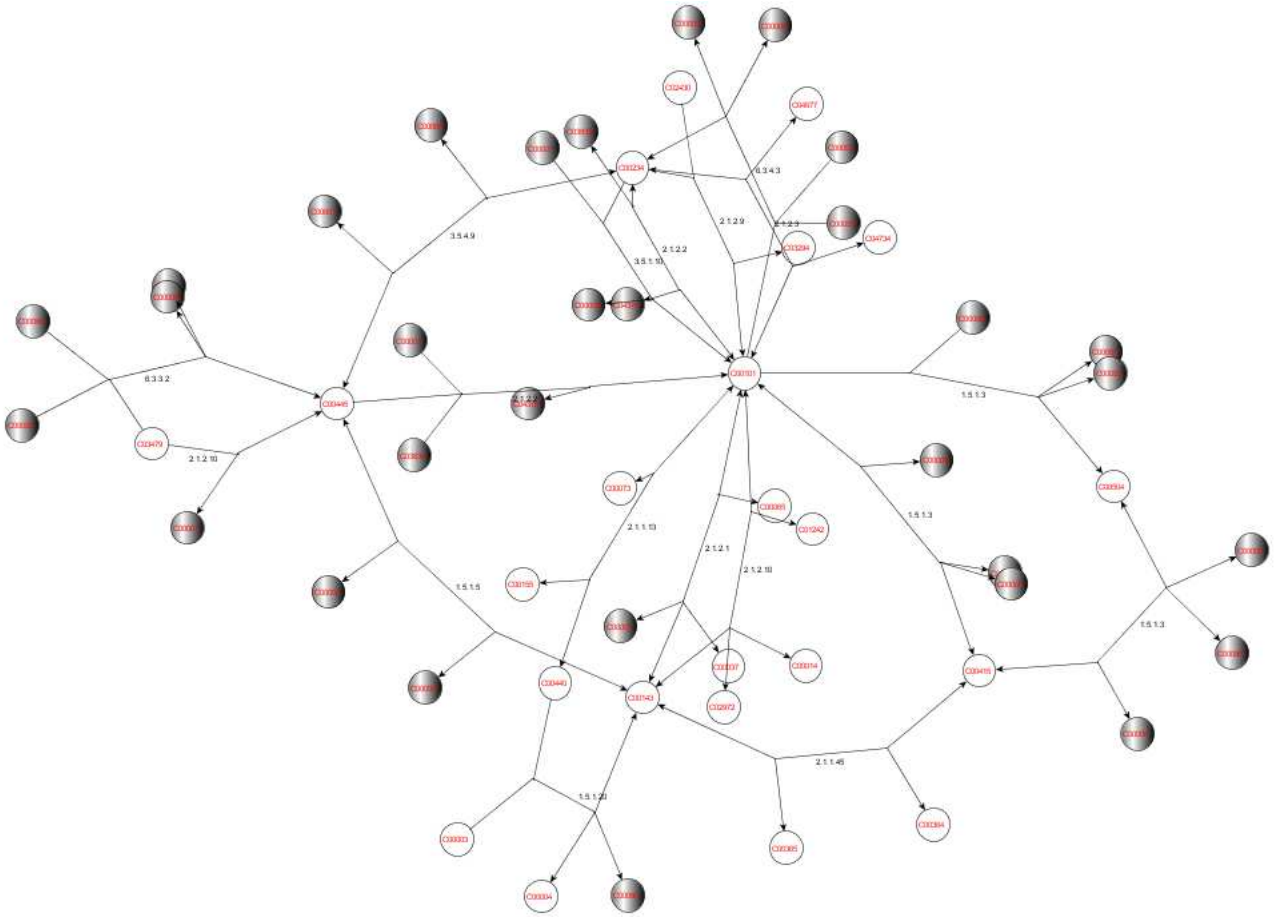


Figure 5.28: *Drawing of One carbon pool by folate of Vibrio fischeri with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned. The pathway has 18 reactions and 33 compounds.*

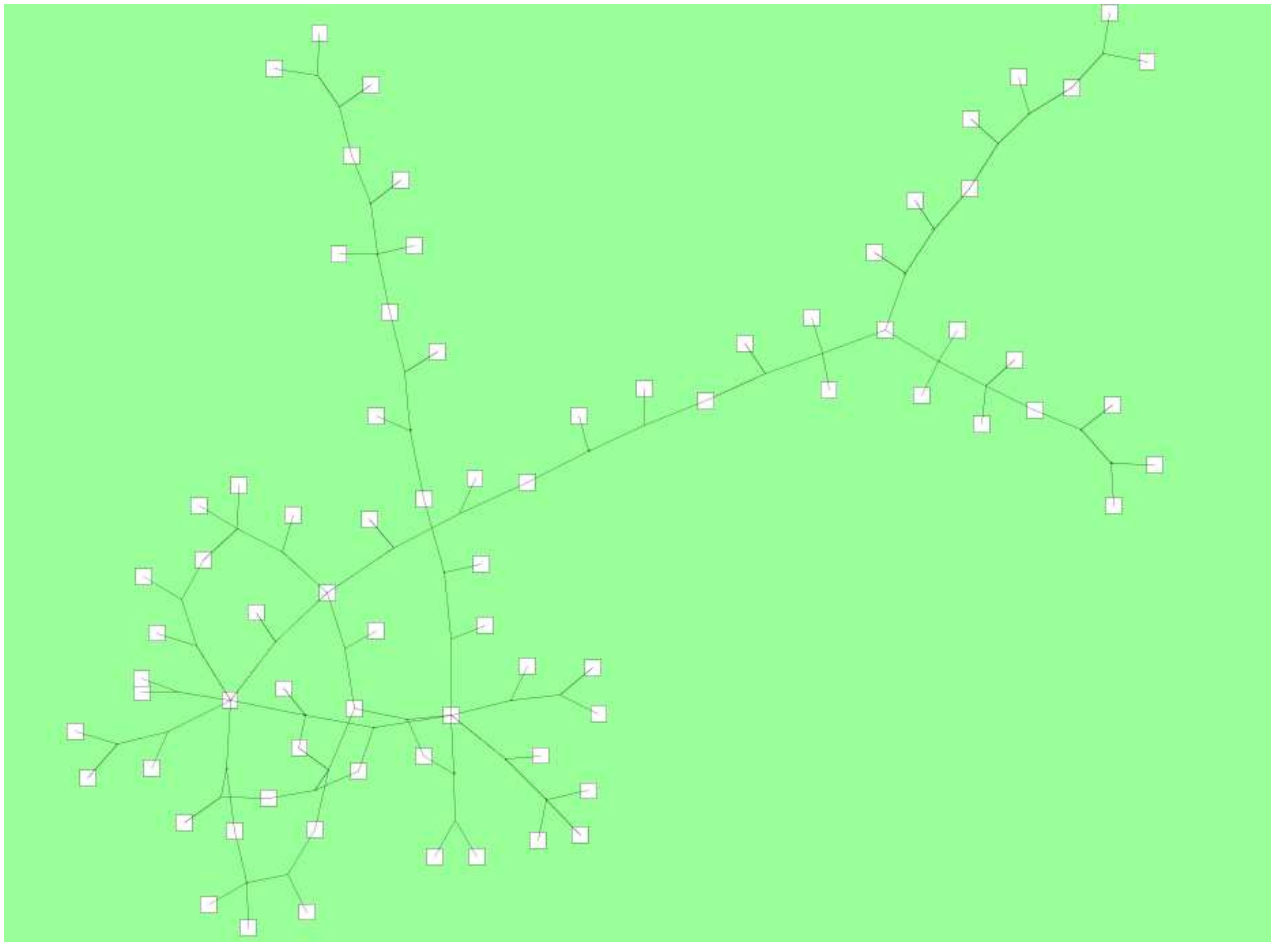


Figure 5.29: *Drawing of Glycine, serine and threonine metabolism of Vibrio fischeri with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned.*

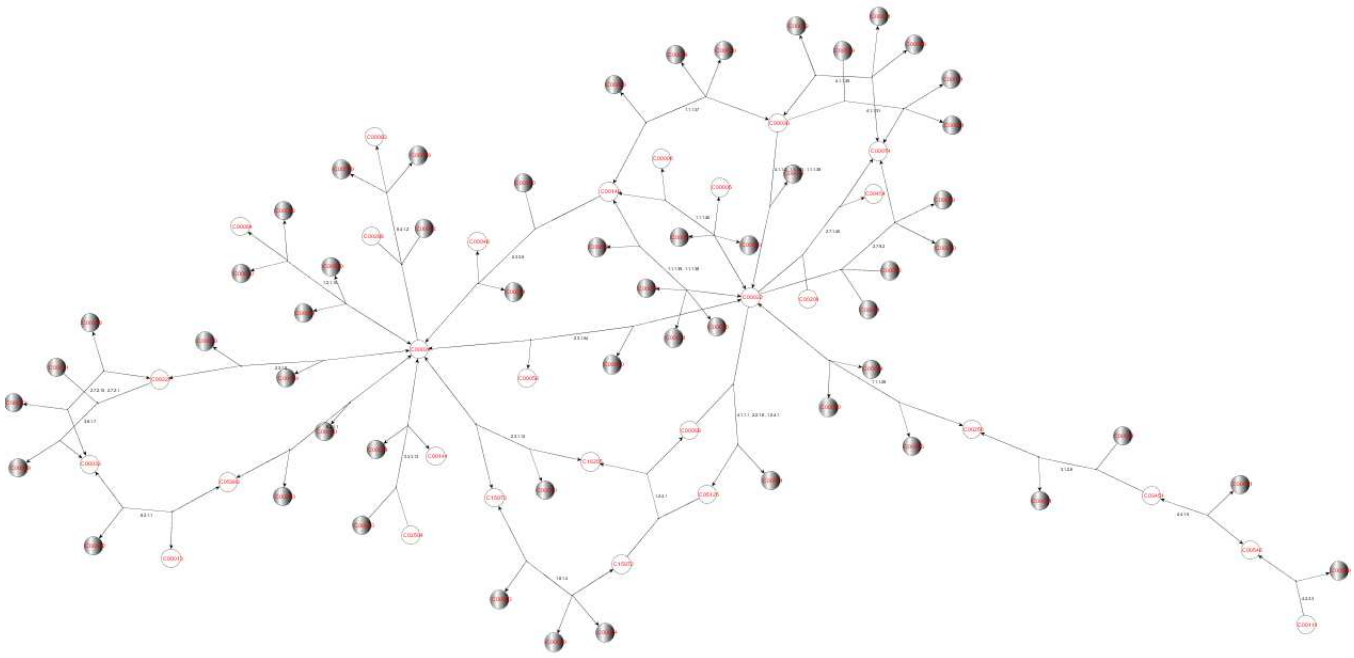


Figure 5.30: *Drawing of Pyruvate metabolism of Vibrio fischeri with organic (classic) layout algorithm using* ***VisBolic****. The grey nodes represent cofactors that have been cloned.*

Chapter 6

Conclusions and Future Directions

The field of biological data visualization currently faces three major challenges: ever increasing quantity of data to be visualized and analysed, integration of heterogeneous -and usually incompatible- data and the representation of multiple connections between nodes with heterogeneous biological meaning.

Data associated with biochemical pathways are stored in many data repositories, with different access mechanisms, formats and structures. In order to exploit the diversity of the data it is necessary to integrate them in a comprehensive manner. The development of automatic parsers for the integration of different data sources is something that has being addressed by many groups [63, 53, 70]. These parsers provide an important support, but in many cases the actual data integration and data curation has be carried out by biologists or biochemists, i.e., people with knowledge in the field who can infer the semantics of the data. This lack of integration and interoperability gave rise to numerous bio-informatics tools (for a comparison of tools see Appendix B). There is a strong need for user-friendly systems allowing the user to handle and combine large and diverse datasets from different sources, generating new insights from them.

The use of an ontology may serve as the basis for the data integration and the automatic extraction of information about the biochemical pathways from free text sources. Such an ontology must be able to describe most of the biological phenomena and complex relationships amongst objects, while maintaining comprehensibility. These descriptions are cumbersome or even not possible under conventional (relational, object-oriented or object-relational) database modelling approaches. Having an ontology expressed as logical facts and having the data expressed in the same way, allows the users to do reasoning over the data, verification of its validity and deduction of facts based on other facts. In [64] an ontology for biochemical pathways is presented. The ontology acts as the basis for the generation of a database, allowing the definition of complex queries and complex data representation.

Also the use of a standardized format for dataset storing may help the process of data integration. Computer science has identified Extensible Markup Language (XML) as an appropriate format for data visualization because it is readable by both, computers and humans. XML stores information in the form of hierarchical tree structures, which allows fast and efficient searching. Many open-source XML-based languages, most notably BioPAX ([22, 1]), SBML ([42, 7]), PSI-MI and MIM [5] exist, containing various levels of complexity and specificity. For an evaluation of different languages, see [71]. Also the various formats of RDF Schemas (e.g. RDF/XML) could be used as an exchange format. RDF Schema is a set of classes with certain properties using the RDF extensible knowledge representation language, providing basic elements for the description of ontologies [1]. The use of a standardized format will help the interoperability between tools, without requiring reprogramming nor understanding of the file format itself.

The modelling should not be limited by the representation of concepts and relationships of

[1]The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model.

direct interest. For example, when talking about a reaction we refer to it not only in terms of its components (reactants and products), but also as an event in time and space, something that takes some time to occur and that takes place at a certain location. This view of a reaction to be seen as spatio-temporal event is normally not reflected in the data model. The third dimension could offer an opportunity to fill this crucial gap of representation of time. Currently, most network tools do not attempt to visualize time series data and thus only produce a static snapshot of all the interactions happening in dynamic systems. Introducing the parameter time as an extra dimension into network visualization tools would thus achieve a more complete picture of complex and highly dynamic biological systems.

Despite the wealth of existing algorithms, the layout problem still remains one of the crucial bottlenecks in network visualization. Faster and more efficient algorithms are needed to bring, especially large-scale networks, into a form that can be easily understood and interpreted by the human brain. One way to circumvent the problem is clustering and graph theory algorithms that can handle large networks. The algorithms should try to minimize both the general and the domain aesthetic criteria. Using a sophisticated implementation of a classical graph drawing algorithm, has the advantage that the application independent criteria are optimized. These criteria include the number of line crossings, the number of bends, the occupied area, the symmetry and the length of edges [26]. The domain specific criteria, are practically the well-established conventions that can be found in relevant biochemistry textbooks:

- The main direction of pathways is accentuated.
- Relevant sub-graphs are placed close to the center of the drawing.
- Reactants and products of a reaction are closely separated.
- Cofactors are placed out of the main path close to the reaction with curved arrows.

- The main compounds lie along the backbone of the pathway.
- Enzymes names are drawn next to the reaction arrow.
- The direction of the pathway should be clearly shown.

Moreover, there is a need for inserting constraints in existing graph drawing algorithms or developing new for hypergraph drawing, since the simulating of hyperedges by inserting dummy vertices does not always give optimal results. The current algorithms treat them like normal nodes, producing sometimes not readable layouts. Finally, the research should be focused on the graphical comparison of metabolic pathways and networks of different organisms.

To adequately address each of these issues, active cooperation will be required between a variety research fields including graph drawing, information visualization, network analysis and of course biology. Only a multidisciplinary approach may lead to to significant results, which implies the merging of biological knowledge and methodologies coming from many and diverse fields. It is unfortunate to witness the number of new tools designed for biology, while have similar in other research areas. For example Pajek, is a general visualization and analysis tool that has limited use with biological networks. This waste of resources is caused by a simple lack of communication between fields. For instance, graph drawing researchers, ignorant of emerging biological models, tend to tackle layout problems of little use in biology or, at best, express solutions to relevant problems in a way that is inaccessible to biologists. A biologically informed graph drawing community, however, would be capable of generating clever ideas for optimization criteria that produce layouts more easily interpreted by biologists.

The key to success of this endeavour is the involvement of all these communities towards standards, open access software, and distributed development. Tools developed in this environment will attract large user communities, avoid duplication of effort and ultimately lead the way towards the goals of systems biology.

Bibliography

[1] Biological pathway exchange. http://www.biopax.org/.

[2] Jfreechart project. http://www.jfree.org/jfreechart/.

[3] Kegg: Kyoto encyclopedia of genes and genomes. http://www.genome.jp/kegg/.

[4] Kuratowski's and wagner's theorems. http://en.wikipedia.org/wiki/Planar_graph#Kuratowski.27s_and_Wagner.27s_theorems.

[5] Molecular interaction maps. http://discover.nci.nih.gov/mim/.

[6] Nomenclature committee of the international union of biochemistry and molecular biology (nc-iubmb). http://www.chem.qmul.ac.uk/iubmb/enzyme/rules.html.

[7] The systems biology markup language. http://sbml.org/Main_Page.

[8] yfiles - java graph layout and visualization library. www.yworks.com/en/.

[9] yfiles for java developer's guide. http://docs.yworks.com/yfiles/doc/developers-guide/index.html.

[10] M. Albrecht, A. Kerren, K. Klein, O. Kohlbacher, P. Mutzel, W. Paul, F. Schreiber, and M. Wybrow. On open problems in biological network visualization. In E. David and

G. Emden, editors, *Graph Drawing*, Lecture Notes in Computer Science, pages 256–267. Springer Berlin, 2010.

[11] G. Ausiello. Directed hypergraphs: Data structures and applications. In *CAAP*, number 295-303, 1988.

[12] G. Ausiello, P. G. Franciosa, and D. Frigioni. Directed hypergraphs: Problems, algorithmic results, and a novel decremental approach. In *ICTCS*, number 312-327, 2001.

[13] G. Ausiello, G. F. Italiano, and U. Nanni. Hypergraph traversal revisited: Cost measures and dynamic algorithms. In *Lecture Notes in Computer Science*, number 1-16, 1998.

[14] M. Baitaluk, M. Sedova, A. Ray, and A. Gupta. Biologicalnetworks: visualization and analysis tool for systems biology. *Nucleic Acids Research*, 34(suppl 2):W466–471, 2006.

[15] A.-L. Barabasi and Z. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[16] S. Basmanova, T. Gaasterl, I. Goryanin, Y. Gretchkin, N. Maltsev, V. Nenashev, R. Overbeek, E. Panyushkina, L. Pronevitch, E. Selkov, E. Selkov, and I. Yunus. The metabolic pathway collection from emp: The enzymes and metabolic pathways database. *Nucleic Acids Research*, 24:26–28, 1996.

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

[18] F. Bertault and P. Eades. Drawing hypergraphs in the subset standard. In *Proceedings of the 8th International Symposium on Graph Drawing*, volume 1984. Springer Verlag, 2000.

[19] U. Brandes, T. Dwyer, and F. Schreiber. Visualizing related metabolic pathways in two and a half dimensions. In L. Giuseppe, editor, *Graph Drawing*, volume 2912 of *Lecture Notes in Computer Science*, pages 111–122. Springer, 2003.

[20] V. Claus, H. Ehrig, and G. Rozenberg, editors. *Graph-Grammars and Their Application to Computer Science and Biology*, 1979.

[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd revised edition, 2001.

[22] E. Demir. The biopax community standard for pathway data sharing. *Nature Biotechnology*, 28(9):935–942, 2010.

[23] E. Demir, O. Babur, U. Dogrusöz, A. Gürsoy, G. Nisanci, R. Çetin Atalay, and M. Ozturk. Patika: an integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics*, 18(7):996–1003, 2002.

[24] Y. Deville, D. Gilbert, J. van Helden, and S. J. Wodak. An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics*, 4(3):246–259, 2003.

[25] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom.*, 4:235–282, 1994.

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

[27] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, 2005.

[28] U. Dogrusoz, E. Z. Erson, E. Giral, E. Demir, O. Babur, A. Cetintas, and R. Colak. Patikaweb: a web interface for analyzing biological pathways through advanced querying and visualization. *Bioinformatics*, 22(3):374–375, 2006.

[29] U. Dogrusoz, E. Giral, A. Cetintas, A. Civrila, and E. Demir. A compound graph layout algorithm for biological pathways. *Lecture Notes in Computer Science*, 3383:442–447, 2005.

[30] L. B. M. Ellis, D. Roe, and L. Wackett. The university of minnesota biocatalysis/biodegradation database: the first decade. *Nucleic Acids Research*, 34(1):D517–521, 2006.

[31] A. J. Enright and C. A. Ouzounis. Biolayout: an automatic graph layout algorithm for similarity visualization. *Bioinformatics*, 17(9):853–854, 2001.

[32] A. Funahashi, Y. Matsuoka, A. Jouraku, H. Kitano, and N. Kikuchi. Celldesigner: a modeling tool for biochemical networks. In *Proceedings of the 38th conference on Winter simulation*, pages 1707–1712, 2006.

[33] A. Funahashi, Y. Matsuoka, A. Jouraku, M. Morohashi, N. Kikuchi, and H. Kitano. Celldesigner 3.5: A versatile modeling tool for biochemical networks. In *Proceedings of the IEEE (2008)*, volume 96, pages 1254–1265. Institute of Electrical and Electronics Engineers, 2008.

[34] A. Funahashi, M. Morohashi, and H. Kitano. Celldesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, 1(5):159–162, 2003.

[35] G. Gallo, G. Longo, and S. Pallottino. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2):177–201, 1993.

[36] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R. D. Appel, and A. Bairoch. Expasy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Research*, 31(13):3784–3788, 2003.

[37] L. Goldovsky, I. Cases, A. J. Enright, and C. A. Ouzounis. Biolayout(java): versatile network visualisation of structural and functional relationships. *Appl Bioinformatics*, 4(1):71–74, 2005.

[38] M. T. Goodrich and R. Tamassia. *Data Structures and Algorithms in Java*. John Wiley & Sons, Inc., New York, NY, USA, 3rd edition, 2003.

[39] S. Goto, Y. Okuno, M. Hattori, T. Nishioka, and M. Kanehisa. Ligand: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Research*, 30(1):402–404, 2002.

[40] Z. Hu, J. Mellor, J. Wu, and C. DeLisi. Visant: an online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, 5(17), 2004.

[41] Z. Hu, J. Mellor, J. Wu, T. Yamada, D. T. Holloway, and C. DeLisi. Visant: data-integrating visual framework for biological networks and modules. *Nucleic Acids Research*, 33(1):352–357, 2005.

[42] M. Hucka. The systems biology markup language sbml: a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[43] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[44] 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(109), 2006.

[45] M. Kanehisa and S. Goto. Kegg: Kyoto encyclopaedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[46] P. D. Karp, S. Paley, and P. Romero. The pathway tools software. *Bioinformatics*, 18(suppl 1):S225–S232, 2002.

[47] P. D. Karp and S. M. Paley. Automated drawing of metabolic pathways. In H. Lim, C. Cantor, and R. Robbins, editors, *Third International Conference on Bioinformatics and Genome Research*, 1994.

[48] P. D. Karp and S. M. Paley. Representations of metabolic knowledge: Pathways. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 203–211. AAAI Press, 1994.

[49] P. D. Karp., M. Riley, M. Saier, I. T. Paulsen, S. M. Paley, and A. Pellegrini-Toole. The ecocyc and metacyc databases. *Nucleic Acids Research*, 28(1):56–59, 2000.

[50] M. Kaufmann, M. van Kreveld, and B. Speckmann. Subdivision drawings of hypergraphs. In I. Tollis and M. Patrignani, editors, *Graph Drawing*, volume 5417 of *Lecture Notes in Computer Science*, pages 396–407. Springer Berlin / Heidelberg, 2009.

[51] S. Killcoyne, G. W. Carter, J. Smith, and J. Boyle. Cytoscape: a community-based framework for network modeling. *Methods in molecular biology*, 563(1):219–239, 2009.

[52] S. Klamt, U.-U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS Comput Biol*, 5(5), May 2009.

[53] B. Kormeier, K. Hippe, P. Arrigo, T. Töpel, S. Janowski, and R. Hofestädt. Reconstruction of biological networks based on life science data integration. *J. Integrative Bioinformatics*, 7(2), 2010.

[54] L. Krishnamurthy, J. Nadeau, G. Ozsoyoglu, M. Ozsoyoglu, G. Schaeffer, M. Tasan, and W. Xu. Pathways database system: an integrated system for biological pathways. *Bioinformatics*, 19(30):930–937, 2003.

[55] 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(Database issue), 2004.

[56] E. Makinen. How to draw a hypergraph. *International Journal of Computer Mathematics*, 34(3):177–185, 1990.

[57] P. Mendes. Advanced visualization of metabolic pathways in pathdb. In *Proc. 8th Conference on Plant and Animal Genome*, 2000.

[58] G. Michal. *Biochemical Pathways: An Atlas of Biochemistry and Molecular Biology*. Wiley-Spektrum, 1998.

[59] A. Mithani, G. M. Preston, and J. Hein. Rahnuma: hypergraph-based tool for metabolic pathway prediction and network comparison. *Bioinformatics*, 25(14):1831–1832, 2009.

[60] C. C. Özturan. On finding hypercycles in chemical reaction networks. *Appl. Math. Lett.*, 21(9):881–884, 2008.

[61] G. A. Pavlopoulos, A.-L. Wegener, and R. Schneider. A survey of visualization tools for biological network analysis. *BioData Mining*, 1:12–22, 2008.

[62] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, 2002.

[63] I. Rojas, L. Bernardi, E. Ratsch, R. Kania, U. Wittig, and J. Saric. A database system for the analysis of biochemical pathways. *In Silico Biology*, 2:7, 2002.

[64] I. Rojas, L. Bernardi, E. Ratsch, R. Kania, U. Wittig, and J. Saric. A database system for the analysis of biochemical pathways. *In Silico Biology*, 2:75–86, 2002.

[65] I. Rojdestvenski. Metabolic pathways in three dimensions. *Bioinformatics*, 19(18):2436–2441, 2003.

[66] I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, and D. Schomburg. Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Research*, 32, 2004.

[67] F. Schreiber. High quality visualization of biochemical pathways in biopath. *Silico Biology*, 2(2):59–73, 2002.

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

[69] M. Sirava, T. Schafer, M. Eiglsperger, M. Kaufmann, O. Kohlbacher, E. Bornberg-Bauer, and H.-P. Lenhof. High quality visualization of biochemical pathways in biopath. *Bioinformatics*, 18(Suppl 2):S219–S230, 2002.

[70] R. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N. W. Paton, C. A. Goble, and A. Brass. Tambis: Transparent access to multiple bioinformatics information sources. *Bioinformatics*, 16(2):184–186, 1999.

[71] L. Stromback and P. Lambrix. Representations of molecular pathways: an evaluation of sbml, psi mi and biopax. *Bioinformatics*, 21(24):4401–4407, 2005.

[72] M. Suderman and M. Hallett. Tools for visually exploring biological networks. *Bioinformatics (Oxford, England)*, 23(20):2651–2659, 2007.

[73] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[74] M. Thakur and R. Tripathi. Complexity of linear connectivity problems in directed hypergraphs. In *FSTTCS 2004*, pages 481–493. Springer, 2004.

[75] M. Thakur and R. Tripathi. Linear connectivity problems in directed hypergraphs. *Theor. Comput. Sci.*, 410(27-29):2592–2618, 2009.

[76] A. Theocharidis, S. van Dongen, A. J. Enright, and T. C. Freeman. Network visualization and analysis of gene expression data using biolayout express(3d). *Nature protocols*, 4(10):1535–1550, 2009.

[77] K. F. Tipton. Nomenclature committee of the international union of biochemistry and molecular biology (nc-iubmb). enzyme nomenclature. recommendations 1992. supplement: Corrections and additions. *European Journal of biochemistry / FEBS*, 223(1):1–5, 1994.

[78] T. Walsh. Hypermaps versus bipartite maps. *Journal of Combinatorial Theory, Series B*, 18(2):155–163, 1975.

[79] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[80] U. Wittig and A. De Beuckelaer. Analysis and comparison of metabolic pathway databases. *Briefings in Bioinformatics*, 2(2):126–142, 2001.

Appendix A

VisBolic Manual

The VisBolic tool has been created for the analysis and visualization of metabolic pathways and networks. It acts as a database browser, as an analyser and as a viewer, offering a familiar and a user friendly interface by which pathways can be constructed automatically from information stored within database. The tool has been designed using an object - oriented approach and is being implemented in Java, so as to provide for platform independence and maximum extendibility due to the ever-growing functionality offered by the language. VisBolic is based on the yFiles Graph library [8] for the basic graph modelling and editing and on the JFreeChart [2] library for the display of distributions as histograms.

The VisBolic is a sophisticated tool that allows to conveniently load and analyse metabolic pathways and networks. Metabolic pathways can be simultaneously displayed and edited in multiple so-called editor windows. For each network an editor window of its own is opened which is shown central in VisBolic application window. The editor window contains the workspace, which dominates the screen and represents the area in which a user views and manipulates the current pathway of interest, and the information table, which presents the elements related to the current pathway. Each editor window has a tab that shows the name of the displayed

Figure A.1: *VisBolic tool logo*

model. By clicking on the tab, the editor window becomes the active editor window. The interface consists also of a menu-bar and a tool-bar. The menu and the tool-bar provide the mechanisms by which the various functionalities of the program can be applied. The main screen of the VisBolic tool is given in Figure A.2.

A.1 Editor - Edit Mode

This mode allows for manipulating a model that is shown in the editor. The following actions can be performed by using the mouse or the keyboard.

Selecting and deselecting nodes: Click on a node to select it. A selected node will have a different appearance than an unselected one. Click on any location in the editor not occupied by a node to deselect all nodes.

By left-dragging the mouse, starting from an unoccupied location in the editor, a selection box will open up. By dragging the mouse around one can change the bounds of the box. Releasing

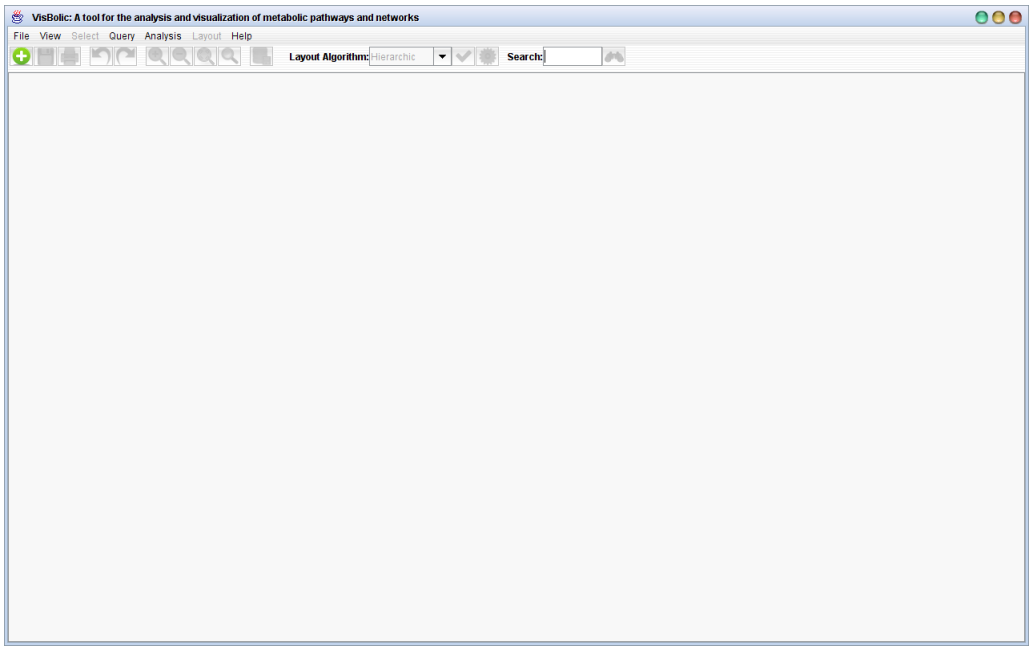


Figure A.2: *VisBolic main screen*

the mouse button will select all nodes that lie within the selection box. All objects outside the box will be deselected.

Opening up a selection box with either the SHIFT key held down adds the selection box nodes to the set of previously selected nodes.

Selecting and deselecting edges: Select an edge by left-clicking on it. Selecting more than one edge requires two steps: first select one edge as described above, then open up a selection box with either the SHIFT key held down to add more edges to the selection. To deselect all edges click on an unoccupied location in the editor.

Move nodes: Select the nodes that you want to move around. Then press the left mouse button over one of the selected nodes and start dragging the mouse. The selected nodes will now move along with the mouse cursor. The nodes will stop moving when the mouse button is

released

Creating bends: Start left-dragging the mouse on an edge. This will insert a new bend on that edge at the pressed location and moves the bend along with the mouse cursor.

Selecting and deselecting bends: Bends are selected the same way as nodes. Either click on a bend or select multiple bends by opening up a selection box with or without either the SHIFT key held down. To deselect all bends click on an unoccupied location in the editor.

Moving bends: First select the bends that you want to move. Then start left-dragging the mouse on one of the selected bends. This will move all selected bends along with the mouse cursor. Releasing the mouse button again stops moving bends.

Context-sensitive menus: Click the right mouse button to bring up a context-sensitive pop-up menu for elements in the editor window. The context is determined by the selection state of the elements.

Selecting and deselecting labels: Labels are selected by clicking on the label text. To deselect a label click on an unoccupied location in the editor.

Moving edge-labels: A label can be moved to another position by pressing the left mouse button on it and then by dragging the label around. While dragging so-called label candidate boxes will appear. These boxes mark the positions that are available for the label. The set of available label positions depends on the label model that is associated which the label. When the mouse button gets released, the label will automatically snap to the closest available label position.

A.2 Editor - Navigational Mode

This mode allows for navigating more conveniently through the diagram being displayed. It does not allow editing of the graph.

Moving the view: The current view of the editor window can be changed by clicking on the arrows on the left of the window. The view will follow the direction depicted by the corresponding arrow.

A.3 The menu-bar

The menu-bar has the following entries File, View, Select, Query, Analysis, Layout and Help.

A.3.1 File menu

Import a model: Opens a dialog, where the user can select a model. A new editor window is used for this model. Models are inserted only from the VisBolic database.

Save as GraphML: Save the current selected model in GraphML format (.graphml). The GraphML format is an XML-based graph format that stores structural information of the graph as well as graphical information for the graph's elements. This format can be used as an interchange format.

Export: Exports the current selected model to a file. This lets the user specify the file name and file format to be used. VisBolic supports different export file formats. Note that models cannot be loaded from exported files.

- Standard PNG Image Writer '.png'.

- Joint Experts Picture Group Image Format '.jpg'
- Graphics Interchange Format '.gif'
- Portable Document Format '.pdf'

The above options will export a screen shot of the current view, using the current size of the image.

Print Preview: Opens up a print preview dialog window which enables the user to preview the print results. Within the preview panel it is also possible to change all relevant print settings and immediately see their effects.

Print: Opens a dialog which enables the user to print either the whole graph or the visible contents of the current view only. The size of the output can be controlled by either specifying a scaling factor to be applied before printing the graph or by specifying on how many page rows and columns the output should fit. When choosing the latter option the scaling factor will be adjusted automatically so that the graph nicely fits on the poster.

Close: Close the current selected editor.

Close All: Close all open editors.

Exit: Quits the program.

A.3.2 View menu

Show/Hide Toolbar: Show (if it is hide) or hide (if it is shown) the tool-bar from the application.

Show/Hide Information Table: Show or hide the information table from the current selected editor.

Show/Hide Grid Lines: Show or hide grid lines from the current selected editor.

Change background color: Change the background of the current selected editor.

Reset background color: Reset the background color of the current selected editor. The default color is white.

Zoom In: Zooms into the current selected view one step.

Zoom Out: Zooms out of the current view one step.

Reset Zoom: Resets the zoom level to 100

Fit Zoom: Zooms and moves current view, so that the whole network gets visible.

A.3.3 Select menu

Select all compounds: Select all compounds of the current selected model.

Select all reactions: Select all reactions of the current selected model.

Select all: Select all elements of the current selected model.

Deselect all: Deselect all selected elements of the current selected model.

Inverse Selection: Select all unselected elements of the current selected model.

A.3.4 Query menu

Make a Query: Opens a dialog for querying the data. The user first chooses the element and then the element's attribute for searching. The results of the query are presented in a new window. The user can navigate from one window displaying information about a particular element, e.g. a reaction, to another window with information about a related object, e.g. an enzyme that catalyses the reaction. The available attributes per element is presented in table A.1. The information presented by query execution is shown in table A.2.

Hierarchies for Compounds: Holds the list of different compound - hierarchies based on specific classifications.

Hierarchies for Reactions: Holds the list of different reaction - hierarchies based on specific classifications.

Hierarchies for Organisms: Holds the list of different organism - hierarchies based on specific classifications.

Hierarchies for Processes: Opens a dialog with the process - hierarchy presented in a tree view.

The user can search for an element's hierarchy based on its ID or its name. During typing, all the element nodes, which their IDs or names start with the current typed ID or name, are selected. The possible classifications for each element are presented in table A.3.

A.3.5 Analysis menu

General Statistics: Opens a window with general statistics over VisBolic database.

Element	**Queries based on:**
Compounds	ID Name Formula
Enzymes	ID Name Class ID
Reactions	ID Name Equation Definition
Processes	ID Name
Organisms	ID Name
Pathways	ID

Table A.1: *Metabolism's elements and their attributes for querying*

Object	Attributes presented
Compounds	Compound ID; Names; Formula; Mass; Chemical Structure; Image of Chemical Structure; Hierarchy; URL link to KEGG database; URL link to mol file; List of reactions the compound participates in; List of processes the compound participates in; List of enzymes the compound is related to; Remarks
Reactions	Reaction ID; Names; Definition; Definition with chemical structures; Diagrammatic representation; Hierarchy; URL link to KEGG database; List of enzyme(s) catalyse the reaction; List of processes the reaction participates in
Enzymes	EC code; Names; Enzyme Class; URL link to KEGG database; List of the reactions catalysed by the enzyme; List of genes regulate the enzyme; Remarks
Organisms	Organism ID; Name; Hierarchy; List of processes the organism has
Process	Process ID; Name; Hierarchy; List of organisms the process occurs to
Pathway	PathwayID; Organism Name; Process Name; List of reactions the pathway has

Table A.2: *Information presented by query execution*

Object	**Classifications**
Compounds	Compounds with biological roles Lipids Phytochemical Compounds Bioactive peptides Endocrine disrupting compounds
Reactions	IUBMB reaction hierarchy Enzymatic reactions Glycosyltransferase reactions
Organisms	Organisms (global classification) Animal classification Plant classification Photosynthetic organisms
Processes	Processes (global classification)

Table A.3: *Classification of metabolism's elements*

Pathway/Network Info: Opens a window with basic information about the current selected model.

Algorithms: Holds the list of different algorithms that can be executed for the current selected model (see table A.4).

Distributions: Holds the list of different distributions that can be computed for the current selected model. The possible distribution are:

- Hypernode Degree Distribution
- Average Clustering Coefficient Distribution
- Average Neighbourhood Connectivity Distribution
- Shortest Path Length Distribution
- Path Length Distribution
- Shared Neighbours Distribution
- Topological Coefficients

Centralities: Opens a dialog that presents the computed centralities (betweenness and closeness centrality) for the current selected model.

Clear Results: Clear the contents of the text output area of the current selected model.

Algorithms	**Description**
Connected Components	Finds the connected components of the network. The connected components are presented in the text output area of the current selected editor window.
Strongly Connected Components	Finds the strongly connected components of the network. The strongly connected components are presented in the text output area of the current selected editor window.
Biconnected Components	Finds the biconnected components of the network. The biconnected components are presented in the text output area of the current selected editor window.
Breadth First Search	If a start hypernode is specified, only those hypernodes that can be reached from the start hypernode are visited. If a start hypernode is not specified, the algorithm chooses a start hypernode and traverses the entire network, choosing a new start hypernode for each sub-network that it traverses. The distance of each hypernode from the start hypernode are presented in the text output area of the current selected editor window.

Algorithms	**Description**
Depth First Search	If a start hypernode is specified, only those hypernodes that can be reached from the start hypernode are visited. If a start hypernode is not specified, the algorithm chooses a start hypernode and traverses the entire network, choosing a new start hypernode for each sub-network that it traverses. The timestamps of each hypernode are presented in the text output area of the current selected editor window.
Topological Sorting	"The network has cycles" is printed in the text output area of the current selected editor window, if there is a cycle in the network. In this case, the nodes cannot be sorted topologically. If the network has no cycles, the order is presented in the text output area.
Shortest Paths	The shortest path distances between every pair of hypernodes are presented in a tabular format. Double clicking in a cell, the corresponding shortest path is presented in the text output area of the current selected editor window.

Algorithms	**Description**
Path exists	The algorithm checks whether there is a hyperpath from a specified start hypernode to a specified finish hypernode. If a hyperpath was found, then it is printed in text output area of the current selected editor window. Otherwise, "A hyperpath does not exist." is printed in the text output area.
Unique path exist	The algorithm checks whether there is exactly one simple hyperpath from a specified start hypernode to a specified finish hypernode. If a unique hyperpath was found, then it is printed in text output area of the current selected editor window. If there is no such hyperpath, "Hyperpath is not unique." is printed in the text output area.
Cycle Detection	The algorithm checks whether a network contains cycles. If it contains, a list of all cycles are presented in text output area, otherwise "The network is acyclic" is printed.

Table A.4: *Algorithms provided by the VisBolic tool for the analysis of metabolic pathways and networks*

A.3.6 Layout menu

The layout menu provides access to the powerful layout algorithms that are part of the yFiles API.

Hierarchical: Runs the classical hierarchic algorithm to the current selected network. This is a layout style that portraits the precedence relation of directed graphs. It is ideal for many application areas, such as bio-informatics, database modeling, software engineering. Use hierarchical layout style to highlight the main direction or flow within a directed graph. Cyclic dependencies of nodes will be automatically detected and resolved. Nodes will be placed in hierarchically arranged layers. Additionally the ordering of the nodes within each layer is chosen in such a way that the number of line (or edge) crossings is small.

Orthogonal (classic): Runs the classic orthogonal layout algorithm to the current selected network. It is a multi-purpose algorithm for undirected graphs. It produces clear representations of complex networks and is especially fit for application areas such as knowledge representation, software engineering, database schema. The algorithm is well suited for medium sized sparse graphs. It produces compact drawings with no node overlaps, few crossings, and few bends. All edges will be routed in an orthogonal style, i.e. only vertical and horizontal line segments will be used.

Orthogonal (compact): This layout algorithm is a specialized variant of the orthogonal (classic) layout method. It uses a divide-and-conquer approach to achieve compact orthogonal layouts that try to respect a specified aspect ratio. The basic idea is to calculate graph partitions which can be laid out according to preferred aspect ratio.

Organic (classic): Runs the organic layout algorithm to the current selected network. It produces clear representations of complex networks and is especially fit for application areas such as bio-informatics, www visualization, knowledge representation. The algorithm is based on the force directed layout paradigm. When laying out a graph nodes are considered to be physical objects with mutually repulsive forces, like protons or electrons. The connections between nodes also follow the physical analogy and are considered to be metal springs attached to pairs of nodes. These springs produce repulsive or attractive forces between their endpoints if they are too short or too long. The algorithm simulates these physical forces and rearranges the positions of the nodes in such a way that the sum of the forces emitted by the nodes and the edges reaches a (local) minimum. Resulting layouts often expose the inherent symmetric and clustered structure of a graph, a well-balanced distribution of nodes, and few edge crossings.

Organic (smart): Runs the smart organic algorithm to the current selected network. It is an improved variant of the organic layout. It combines different techniques and layout algorithms depending on the user's demands. Its strength is its ability to control the ratio of quality versus running time, as well as its ability to produce organic layouts while guaranteeing the absence of node overlaps.

Circular: Runs the circular layout algorithm to the current selected network. It is an algorithm that portraits interconnected ring and star topologies and is excellent for applications in social networking, www visualization. Circular produces layouts that emphasize group and tree structures within a network. It creates node partitions by analysing the connectivity structure of the network, and arranges the partitions as separate circles or disks. These circles are arranged in a radial tree layout fashion.

Settings: Shows a dialog with the settings for the current selected layout algorithm. The details of the settings can be found in yFiles Documentation [9].

A.3.7 Help

About: Displays version and legal information.

A.4 The tool-bar

The additional functionalities provided by the tool-bar are:

Redo: Undoes the last modification to the current selected view.

Undo: Redoes the last undo step to the current selected view.

Magnifier: Shows a magnifier for a more detailed view to specific locations of the current selected network.

Save Data: Save the data presented in the information table of the current selected editor window in '.xls' format .

Search: The search component let the user find elements of the current selected model, based on the elements IDs.

A.5 Future Work

The VisBolic tool could be undergone many improvements. Firstly, a user-friendly interface has to be developed for updating the VisBolic database. The classical user interfaces for manip-

ulating databases require knowledge about SQL queries and relational databases, a knowledge that many biologists usually do not have. Also, methods for ranking the answers, when a user makes a query, are necessary. Given that the metabolic knowledge is enormous, an information retrieval system would be extremely helpful. Furthermore, the database could be enriched with additional information about metabolism, i.e. genes, enzyme inhibitors etc. Besides, the possibility of loading a model from a file, not only from the database, should be available in next versions. The import formats should be standardized, xml-based helping the interoperability between tools. Finally, the implementation of additional metrics and additional functionalities, e.g. deactivation of hypernodes or network editing, should be carried out.

Appendix B

A comparison of visualization tools

Several tools have been developed to represent, visualize, and analyse metabolic networks and data. The wealth of existing visualization tools makes an exhaustive collection and in-depth discussion of all available software tools impossible. We present a selection of several network viewers, which are broadly applicable. We will focus on tools that produce automatic network layout and we will omit more general-purpose tools like Pajek.

Reference Features (see table B.1)

- **Publications:** Publications describing the application.
- **Year:** Year the software first appeared.
- **Most Recent Release:** Year of the most recent release. Gives an indication of how actively the tool is being developed.
- **Tried:** It was not possible to try all tools because in same cases the tool is no longer available.

Programs	**Website**	**Publications**	**Year**	**Recent Release**	**Tried**	License
VisBolic	•	(not yet)	2009	2011	Yes!	free to use
BioLayout	http://www.biolayout.org/	[31, 37, 76]	2001	2011	Yes	free for academic use
BiologicalNetworks	http://biologicalnetworks.net/	[14]	2006	2010	Yes	free to use
CellDesigner	http://www.celldesigner.org/	[34, 33, 32]	2003	2010	Yes	free to use
Cytoscape	http://www.cytoscape.org/	[68, 51]	2003	2010	Yes	open source
Pathway Tools	http://bioinformatics.ai.sri.com/ptools/	[46]	2002	2007	No	free for academic use
PatikaWeb	http://www.cs.bilkent.edu.tr/~patikaweb/	[23, 28]	2002	2006	Yes	free to use
Rahnuma	http://portal.stats.ox.ac.uk:8080/rahnuma/	[59]	2009	2009	No response	free for academic use
VANTED	http://vanted.ipk-gatersleben.de/	[44]	2006	2011	Yes	free to use
VisANT	http://visant.bu.edu/	[40, 41]	2004	2010	Yes	open source

Table B.1: *A comparison of reference features of tools for visually exploring metabolic networks*

Programs	**Databases**	**Queries**	**Model**	**Analysis**	**Visualization**
VisBolic	KEGG	Select node/edges by name, Predefined queries on elements of metabolism	Hypergraph based	Connected, biconnected and strongly connected components, DFS, BFS, tolopological sorting algorithm, shortest paths, all-pairs shortest paths algorithm, distributions etc. For a detail description see Appendix A	Hierarchic, orthogonal, organic, smart organic, circular

Programs	**Databases**	**Queries**	**Model**	**Analysis**	**Visualization**
BioLayout	•	Select node by name/class	Graph-based, 3D Model	Network connectivity, degrees, network diameter	Force - directed
BiologicalNetworks	PathSys	Select paths of length k, shortest paths between node sets	Graph-based	Clustering	Circular, spring-embedded, tree, hierarchic, radial, random
CellDesigner	BioModels.net	•	Graph-based	•	Orthogonal, organic, spring embedding, hierarchic, circular, tree, undefined default, several ways to route edges

Programs	**Databases**	**Queries**	**Model**	**Analysis**	**Visualization**
Cytoscape	•	•	Graph-based	Plug-ins with many statistical metrics implemented	Force-directed, circular, hierarchic, grid layout
Pathway Tools	BioCyc	Select node neighbours	Graph-based	•	Algorithms specialized for metabolic pathways based on [47]
PatikaWeb	Patika	Simple or complex queries on elements of metabolism	Graph-based	Select distance-d neighbours	Force-directed based on [29]
Rahnuma	KEGG	Queries focusing on differences between organisms	Hypergraph based	Computation of all possible paths between two metabolites	•

Programs	**Databases**	**Queries**	**Model**	**Analysis**	**Visualization**
VANTED	KEGG, MetaCrop database	•	Graph-based	Shortest-paths and circle detection algorithms	Force-directed, circular, grid, tree layout, random
VisANT	Predictome	•	Graph-based	Computation of degree distribution, average path length and shortest path length between two nodes , algorithm for detection of common motifs	Force-directed

Table B.2: *A comparison of functional features of tools for visually exploring metabolic networks*

Functional Features (see table B.2)

- **Databases:** The main databases for which the software tool has been designed.
- **Queries:** Facilitating for asking questions about the elements of metabolism.
- **Model:** The model that is used for the analysis and representation of networks.
- **Analysis:** Metrics for the analysis that are supported by the software tool.
- **Visualization:** Visualization algorithms that are implemented or supported (if the tool use third-party software) by the tool.

Other specialized tools are Genomes Pathway Browser [54], BioMiner [69], BioPath [67] and PathDB [57]. A detail comparison of visualization tools is presented in [72] and in [61].

The majority of tools presented above support queries on the metabolic network or pathway that is presently displayed. VisBolic tool is the only that supports the execution of queries in the whole database of metabolism. Scientists will benefit from quick electronic access to information about the metabolism supporting their research activities. Besides, VisBolic offers a wide range of statistical metrics for the analysis of underlying networks. The majority of tools are characterized by a lack of analysis features, limited to the computation of shortest path between two metabolites and the computation of degrees and degree distributions. Due to the complexity of the metabolism, there is a need for useful metrics that will help the efficient analysis of networks. Finally, the tools should support hypergraph representation of networks. However, only Rahnuma and VisBolic support such representations. In chapter 5, the readability of the drawings that are produced by VisBolic and by other tools such as Cytoscape, CellDesigner and VANTED can be compared. It is obvious that the representation of metabolic networks using hypergraphs offers more accurate and "clear" drawings.

On the other side, we must admit that many of the tools presented above are not only metabolism-oriented. There are tools that support the analysis and visualization of general biological networks or enable the integrative analysis of biological networks together with transcriptomic, metabolomic and proteomic data. Although the existence of numerous bioinformatics tools, none can capture all the requirements of the metabolism context. The tools usually approach one or more (but never all) aspects of metabolism while neglecting others. However, even these aspects are important for the understanding of metabolism. For example, although CellDesigner offers an integrative analysis of biological networks, a graph-based model is used for the analysis and visualization. On the other hand, although Rahnuma offers a hypergraph based model, it does not support at all visualization feature. In conclusion, although VisBolic is a notable tool for the querying, analysis and visualization of metabolic networks, there are still many features to be improved. These features range from practical, such as the improvement of the graphical interface, to more complicated such as the integration of data from different sources or the integration analysis of various biological networks.