UNIVERSITÄT LEIPZIG
Fakultät für Mathematik und Informatik
Institut für Informatik
Bioinformatik

Evolution of Metabolism in a Graph-Based Toy-Universe

Diplomarbeit

Betreuende Hochschullehrer: Prof. Peter F. Stadler
Prof. Christoph Flamm

Leipzig, February 1, 2008

vorgelegt von
Alexander Ullrich
geb. am 31. März 1982
Studiengang Informatik

Acknowledgement

I want to thank everyone who contributed to my work and supported me throughout my studies.

First of all, my advisor -Peter Stadler- who increased my interest in the area of complex biological networks, inspired me to find new solutions through his ideas and questions on the topic, and also helped me in other situations during my studies. During my time in vienna, my other advisor -Christoph Flamm- who provided me with knowledge and new ideas on a daily basis, was an invaluable help and support. I also want to thank all the people in the TBI-Vienna group as well as the Bioinformatik Leipzig group and, especially, Dill, Konstantin, Lukas, Rainer and Svrci for literature and interesting conversations.

I wish to thank all of my friends and family. Andreas, Heiko and Martin for the many long evenings we spend on homeworks or preparing for exams. I also want to mention here the brave people who read through this thesis. Natallia and Vicky, I very much appreciate your effort in the ”which”hunt, and, I, am, thankful, for, every, CS. In particular, I want to thank Karen for her endless help. Even more than for the proof-reading and the editing of pictures, I am grateful for the daily motivation, support and affection.

Most importantly, I want to thank my family, my sister and my parents, who always believed in me and supported me in every situation. My special thanks to my parents who provided me with a carefree environment, every conceivable support, and the occasional pushing, which enabled me to pursue my studies successfully.

Danksagung

Ich danke allen, die an dieser Arbeit beteiligt waren und mich während meiner Studienzeit und anderweitig unterstützt haben. Dazu zählen die Kollegen in der Bioinformatik-Gruppe in Leipzig und der TBI-Gruppe in Wien. Besonderer Dank geht an dieser Stelle an meine Betreuer Peter Stadler und Christoph Flamm. Weiterhin möchte ich meine Dankbarkeit gegenüber Freunden und Familie zum Ausdruck bringen, denen ich viel verdanke.

Ganz spezieller Dank geht an meine Eltern, die mir immer jede erdenkliche Hilfe erbracht haben, mich sorgenfrei entfalten liessen und ab und an den nötigen Schub versetzt haben.

Contents

List of Figures

List of Tables

1 Introduction

> Life emerged, I suggest, not simple, but complex and whole, and has remained complex and whole ever since–not because of a mysterious elan vital, but thanks to the simple, profound transformation of dead molecules into an organization by which each molecule's formation is catalyzed by some other molecule in the organization. The secret of life, the wellspring of reproduction, is not to be found in the beauty of Watson-Crick pairing, but in the achievement of collective catalytic closure. The roots are deeper than the double helix and are based in chemistry itself. So, in another sense, life–complex, whole, emergent–is simple after all, a natural outgrowth of the world in which we live.
>
> Stuart Kauffman [Kau93]

1.1 Motivation

Life, in the most basic sense, constitutes of interactions between chemical compounds, building complex networks which in turn can be regulated and interacted with. In a living organism such complex networks of interactions of molecules are called metabolism. Living organisms adopt to their environment by means of gradual change of the internal networks and regulations. Since Darwin's "The Origin of Species" [Dar93] we know that populations evolve from generation to generation through natural selection. This is because individuals in a population differ from each other and hence are not equally well adjusted to their environment. Individuals, which are better adapted are more likely to survive and reproduce and thus pass on the properties which make them adjust so well. This leads to well adapted populations in the end. We have a nearly complete knowledge about the process of reproduction. We learned the basic principle and the connection between genotype and phenotype through Mendel's laws. Watson and Crick taught us the building blocks and modern science revealed every bit of our genome. Through advances in technology, we are able to see what is happening inside of us, we can understand what makes organisms suited for the environments they live in. But despite all this knowledge, we might never know with certainty how all these parts came into being and how they developed into their present state. Natural selection is the big piece in the puzzle, but it is not the only one. However, together with the analysis of the properties of living organisms a clearer view on the remaining pieces can be provided.

Metabolic networks are the best studied biochemical networks. We can reconstruct entire metabolisms because we have complete annotated genomes of model organisms at our disposal. Looking at

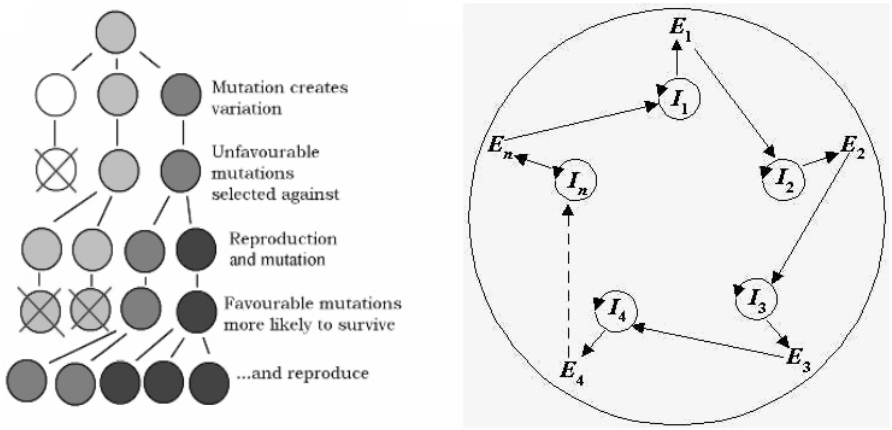


Figure 1: Two principles of evolution: Natural selection and self-organization.

pathways in these networks can in turn be interesting for functional genomics, e.g. gene expression data derived from DNA arrays may be better understood in terms of metabolic components, pathways or sub-networks. Insights about the metabolism of an organism can of course be useful for further biotechnological applications [Lia96], e.g. the determination of pharmaceutical targets, metabolic engineering, changing direction and yield of pathways. Before we can make use of all these applications, it is essential to gain an understanding of the network properties. It is often not enough to look at textbook-like metabolisms, since they do not resemble the actual behavior of these networks. Therefore, we need means to analyze the network's topology, structure, principle, plasticity and modules. Such means exist, e.g. metabolic flux analysis [Sch99][PSVN$^+$99], and thus we are able to make observations similar to the following: there is a large number of diverse enzymes; metabolic networks are small world networks and they contain a number of hub-metabolites. But it also has to be noted that, first, there are still some limitations to the analysis, as we will discuss in chapter 5 Metabolic Flux Analysis, and secondly, the question remains how these and other possible properties emerged and further evolved. The case is especially difficult if we regard properties which cannot be sufficiently explained by looking at a static network image [PNW$^+$03], e.g. robustness or evolvability.

It can be assumed and it is believed by most biologists that all organisms which we can see today are evolved from one common ancestor [KB84]. This ancestor would be a single cell having properties similar to those observed in cells of modern organisms. Since we do not believe this cell to have been spontaneously and magically appeared on the earth's surface, it is fair to suggest that this cell in turn gradually evolved from simpler cells. Many theories on the origin of life and scenarios for the early evolution exist, but actually we cannot say anything

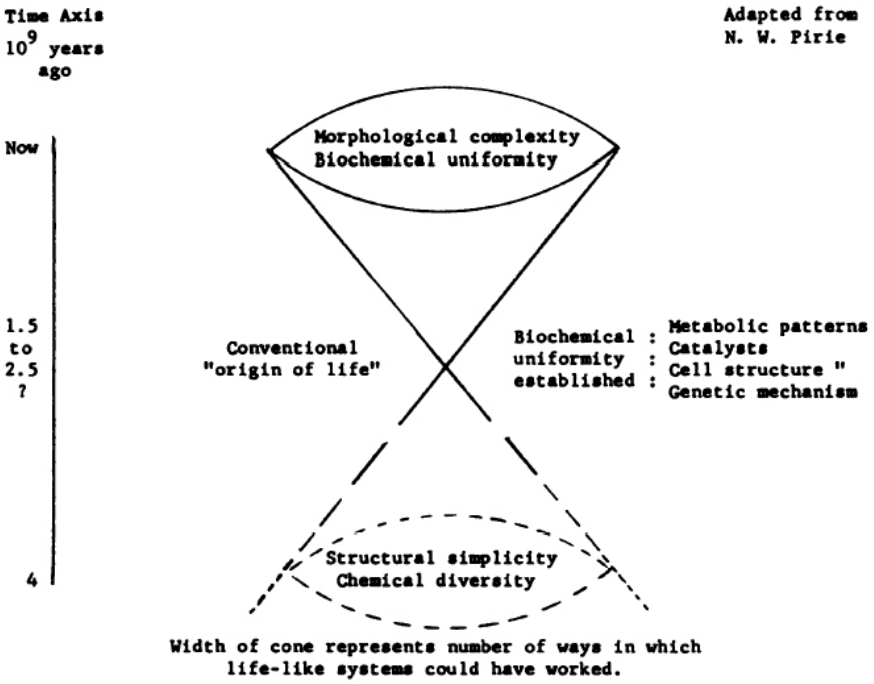


Figure 2: Theory for the evolution of metabolism by Eakin [Eak63]

with certainty until the point of the common ancestor, thus all the available modern molecular techniques will fail to provide a complete account of the emergence of some of the properties we are interested in. One scenario for the evolution of organisms is depicted in figure 2, which ought to illustrate the mentioned problem rather than imply meaning or truth of this scenario [Eak63]. Another problem is the determination of the relations between enzymes or other common complex molecules. According to Dayhoff's definition[Day76], we would have to group all enzymes into approximately 200 superfamilies and moderate guesses are that this number will increase to at least 500. This classification may be very helpful and make sense in terms of the enzyme's function. However, it would suggest that the common ancestor cell would have started with at least 500 enzymes, all unrelated to each other. This being unlikely, it is desirable to come up with theories of how enzymes emerge and evolve.

Models for the simulation of the origin of network properties, as discussed above, do exist and have provided explanations for some of these properties. For example, [DMPRS07] showed that gene-duplication may account for the property of a network to be scale free. So far these models of biological systems use either differential equations [PS05], i.e. enzymes are not modeled as actual chemical entities but only as rates, or they use very abstract artificial chemistries. A more

complex simulation integrating more functional constraints of the metabolism should provide further insights about the metabolism itself, properties of complex networks in general and also their emergence, so that these properties may be reproducible in other applications. For instance, artificial networks are desired to be robust and maybe even evolvable as well.

1.2 Objective

Within this thesis a new model for the simulation of metabolism will be introduced, which can work under different circumstances and in which hypotheses about network structures, function and properties which go beyond those of single components can be tested. Another objective of this work is an implementation of a state-of-the-art method for metabolic flux analysis which can be used as a stand-alone tool as well as in the simulation tool.

The model described in this thesis and used in the simulation tool ought to possess some properties expected from metabolic networks and evolving systems. Thus it has to be designed in such a way that it resembles a system that is able to perform self-organization and show signs of self-modification and maybe even self-shaping. It also has to be statistically tested whether it is evolvable. In order to introduce some more complexity than other approaches of cell simulations, a more sophisticated artificial chemistry has to be integrated. Besides, a realistic model ought to be achieved by using an overall graph-based approach. The model should enable the user to test theories of the emergence of basic properties of complex networks such as yield, robustness, correlation or minimal media, as was shown possible for example by [PS05], explaining the presence of hub-metabolites in metabolic networks. In order to be testable for a broad range of hypotheses, the model must be fairly unbiased and include as few assumptions as necessary. Also we want to be able to answer various metabolic specific questions starting from simple minimal media determination to highly complex questions of the evolution of metabolism in complex biological systems. For example: ”Are thermodynamic workcycles performed?”, ”Do we observe darwinian adaptations?”, ”are the observed features consequences of evolution or of the chemistry?”, ”can failures be anticipated?”, or in other words, whether different modules in the metabolic network evolve without shifting of the environment. Thus we need a flexible model where different parameters such as the environment or the chemistry can be adjusted. Finally, the results of the simulation runs have to be accessible in an appropriate way to analyze the important characteristics of the evolved networks and their path of evolution, e.g. relations among genes, enzymes or individuals and the connection between changes in genotype and changes in phenotype.

The other goal of this work is the implementation of a metabolic flux analysis tool. On one

hand, it should apply all the advances made in this area so far [GK04, UW05a], e.g. memory and time efficiency, and improve them by extended reduction steps. On the other hand, it has to be adaptable to the simulation tool and integrate useful functions for the analysis of the evolution of metabolism. It is important to mention here in particular the possibility of gaining knowledge of single enzyme knockouts without redoing the entire analytical computation.

2 Background

> There is a natural-selection explanation for the fact that what goes up must come down, since -The stuff that didn't come down isn't here anymore!-
>
> Sidney Morgenbesser

2.1 Metabolism

Metabolism can be regarded as the set of chemical processes that create and use energy and thus allow the cell to grow and reproduce. The chemical processes or reactions are catalyzed by enzymes. The substances participating in the chemical reactions are called metabolites. Through metabolism some of these substances are regarded nutritious and some harmful and the cell can respond to its environment by treating the substances accordingly. Metabolism can either generate energy by breaking large molecules (catabolism) or build large molecules like structural proteins, enzymes or nucleic acids (anabolism).

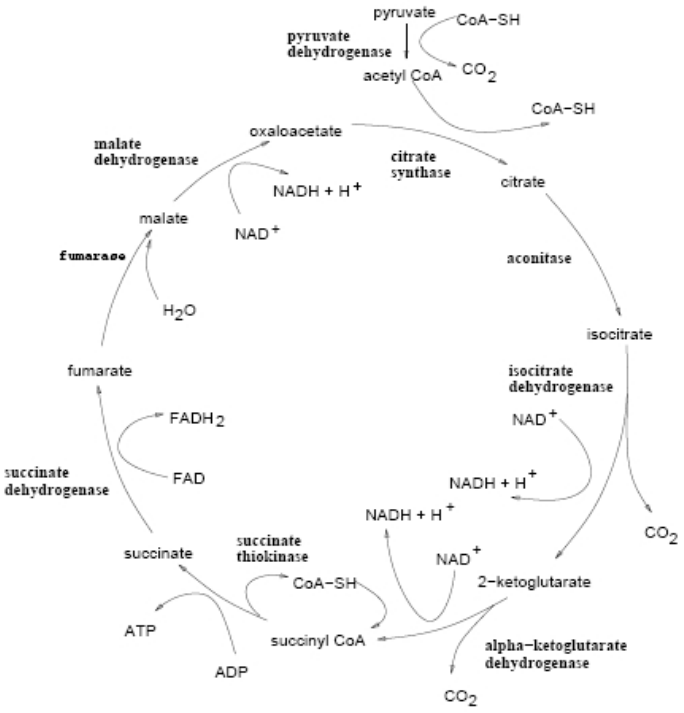


Figure 3: Citric acid cycle [KM94]

Metabolism is commonly analyzed in terms of metabolic pathways[PNW+03], which are a

series of enzymes performing chemical reactions on a metabolite. These pathways are often forked, i.e. more than one enzyme can react with the metabolite and feed back on themselves, i.e. the pathway becomes a cycle. Pathways are also usually intertwined, feeding each other, exchanging metabolites or sharing enzymes, thus creating an entire network. For example the citrate cycle, figure 3, is fed by glycolysis, feeds back and feeds itself respiration.

2.2 Evolution of Metabolism

The evolution of metabolism is the main theme of this thesis, thus we dedicate one section to explain what kind of questions stand behind this topic. Some ideas for the origin of metabolism will be discussed as well as some theories about the emergence of the connectivity of metabolites the way we can observe it. Interesting research work has also been done on the evolution of enzymes and will be presented in the last paragraph.

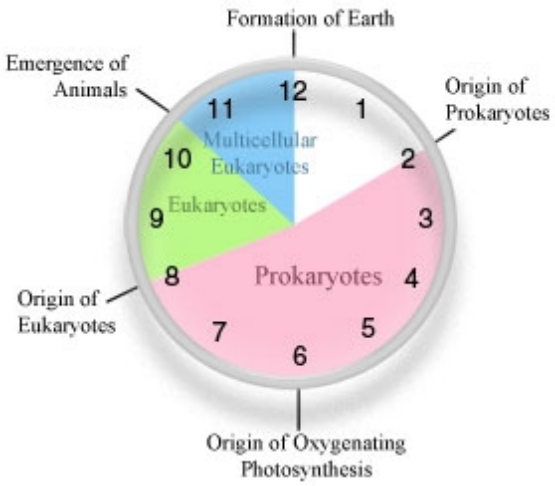


Figure 4: Evolution of life on earth

2.2.1 Origin of Metabolism

As already mentioned, it has been generally accepted that all modern cells originate from one common ancestor cell and in turn this cell evolved from simpler protocells. This leads to the problem that it is not possible to make verified statements about these precursors. Hence, there exist many different theories on the origin of the first organic molecules as well as the first protocell. It is not even known with certainty in which order they occurred in the process of evolution.

All models for the origin of life or metabolism have one thing in common: they are difficult to test and very few experiments exist. Probably, the best known experiment is that of Miller and Urey[Mil53], in which a mixture of methane, water and ammonia was used to build organic molecules out of anorganic ones. However, these experiments can not solve the mystery: first of all, the mixture of gases used in their experiments does not comply with the assumptions about the composition of the atmosphere of the early earth and , secondly, it fails to give an explanation for the polymerization of organic molecules or the emergence of replicating protocells.

It is difficult to group the different approaches because they either try to explain the origin on different levels of abstraction or address only parts of the problem. An important theory about the formation of self-replication stems from Eigen and Schuster[ES79], in which hypercycles are introduced, containing information storing systems, that in turn can produce some kind of enzyme, helping to replicate another information storing system, leading to circular sequence and thus quasi-species. Hence, such a hypercycle, as well as its information storing systems and enzymes, can subsequently evolve through natural selection. This hypercycle theory, however, does not account for the emergence of complex organic molecules needed for the information storing system and the enzymes. Other approaches deal with the question of the complexity and constitution of the environment[Har75], or suggest molecules responsible for function in early organisms and thus subject of evolution. [Eak63], for example, makes a point for co-factors being enriched, accounting for the evolution of metabolism. These co-factors, being precursors of the today known co-factors, possess a function for the metabolism of the protocell and also can be passed down to their descendants, thus fulfilling the requirements to constitute a system on which evolution can engage. Both works are interesting for us because they imply that simple conditions suffice for the occurrence of transformations from anorganic to organic and replicating units.

Contrary to the theory of the ”thick soup” or suggestions that early primitive cells had undeveloped regulatory mechanisms[Jen76], there exist a number of theories assuming the origin in anorganic protocells and the evolution of metabolism, regulation and cell structure to be simultaneous[LK97]. Among these are the liposome hypothesis[DO80], stating that membranes of lipid bilayer predate proteins or nucleic acids, but using their monomers to fulfill some of their functions, the clay theory[Ber67], and the regosome model[Nus78]. According to the clay theory, the surface of the clay is able to concentrate organic molecules and also catalyze their polymerization. It is proposed that the clay crystals may even serve as genetic material[CS66]. Other authors emphasized the importance of different minerals[Wäc88, Har75, RD94]. A different role is assigned to clay in the regosome model, in which regoliths, small clay dust grains, due

to their porosity can serve as compartments, in which chemical processes could be performed and the origin of life could occur. In addition, such regolith grains can concentrate lipids on their surface, forming an entire layer around it. This would add up to a highly potential protocell.

2.2.2 Connectivity of Metabolites

It is known that metabolic networks bear the small-world property[WF01], i.e. its longest path (diameter) is short, it has a power law degree distribution and it contains few highly connected nodes (hub-metabolites). In order to gain insight about the abundance of hub metabolites, Pfeiffer designed an abstract metabolism in which enzymes can evolve[PS05]. Within their simulation enzymes evolved from enzymes of low specificity to highly specified ones. The explanation for the occurrence of hub-metabolites in this scenario then is that during the evolution due to the specialization, some reactions and, consequently, also intermediary metabolites are lost. It is also proposed that the hub-metabolites occur due to selection for growth rate and not solely robustness as was believed before. Some assumptions used in the simulation should be noted: firstly, the metabolic network is exclusively a group transfer network, i.e. metabolites consist only of functional groups and enzymes can only transfer groups from one metabolite to another. Metabolites, thus, can be represented as binary sequences, each position indicating whether a functional group is contained in the metabolite. Other assumptions are that enzymes are monomolecular and the existence of transporters responsible for the uptake and excretion of metabolites. The basic model of the simulation is depicted in Figure 5. The simulation integrates the tradeoff between increased flux and metabolic costs when the dosage of an enzyme increases, e.g. due to duplication and also the tradeoff between specificity and catalytic activity. The assumption here is that if a transporter specializes on one metabolite rather than two, its catalytic rate for this metabolite is four times higher than that for each of the two original metabolites. The assumptions regarding the metabolites and enzymes could be integrated as well into simulations with our model by choosing an appropriate set of metabolites as environment and chemical reactions. However, assumptions toward the tradeoffs are not desired here because they might bias the experiment to a certain outcome.

Two completely different aspects are studied in [SS06] and [Sin]. In the former, it is shown how the connectivity of metabolites can be used to explain essential reactions and determine modules in metabolic networks. It is stated that a reaction can be considered essential if it is connected to two sparsely connected metabolites and in turn modules can be discovered by looking for such essential reactions resembling bridges between modules. In the latter study, the hypothesis is proposed that the degree of a metabolite is proportional to the change in its reaction set compared to other species. They use metabolic networks from organisms, some of which are

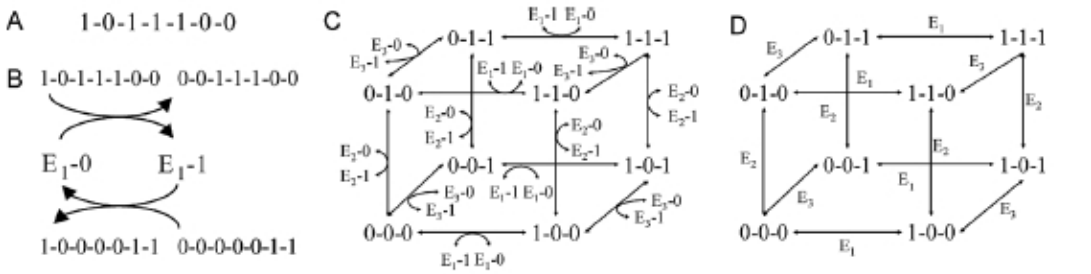


Figure 5: Pfeiffer's simulation model [PS05]. A) Representation of a metabolite as a binary sequence. B) Enzymes represented as half reactions, only group transfer. C) Model of the bimolecular network. D) Model of the monomolecular network.

closely related and others which are evolutionary distant. The same could possibly be done for a simulation of evolving metabolic networks.

2.2.3 Evolution of Enzymes

Another area of interest is the evolution of enzymes, the basic question is here whether a protocell started with a few multifunctional enzymes which then specialized into many highly specific enzymes. One also wants to understand by which means, i.e. biological principles or properties, the respective route in evolution was inevitably chosen. Two of the first models are that of retrograde evolution[Hor45], and patchwork evolution[Jen76]. In retrograde evolution, depletion of metabolites in the environment causes enzyme specification, because gene duplication could possibly create an enzyme that produces the missing metabolite, which in turn also would lead to a coupling of the two enzymes. The patchwork evolution model, as explained by [Jen76], however, proposes that the gene duplication of multifunctional enzymes gives both enzymes the possibility to adopt only part of the functionality of the original enzyme. Both hypothesis are supported and can be explained by looking at metabolic networks of different organisms. Thus, it can be assumed that both bear some truth in them. In fact, network-based approaches like that of [DMPRS07] suggest that both concepts can or rather ought to be combined. Nevertheless, it is believed that patchwork evolution accounts for the biggest part in enzyme evolution. [DMPRS07] also emphasize the role of biochemical coupling of enzymes in explaining the abundance of duplicated genes. Another slightly different explanation is provided by [KB84], which does not rely on gene duplication for specification of enzymes to occur. The presented system involves a set of different multifunctional enzymes, possessing an overlap of functionality. Thus, there is again a situation of multiple realization of a function and consequently mutations and natural selection allow the enzymes to specify without detriment in yield. It should be noted that

[KB84] does not oppose the ideas of those, discussed above, because the overlap of functionality could be accounted for by the relatedness of those enzymes. However, it is not a necessity. Figure 6 illustrates the idea incorporated in all approaches. That is, the duplication of function or genes allows for evolution and specification of enzymes.

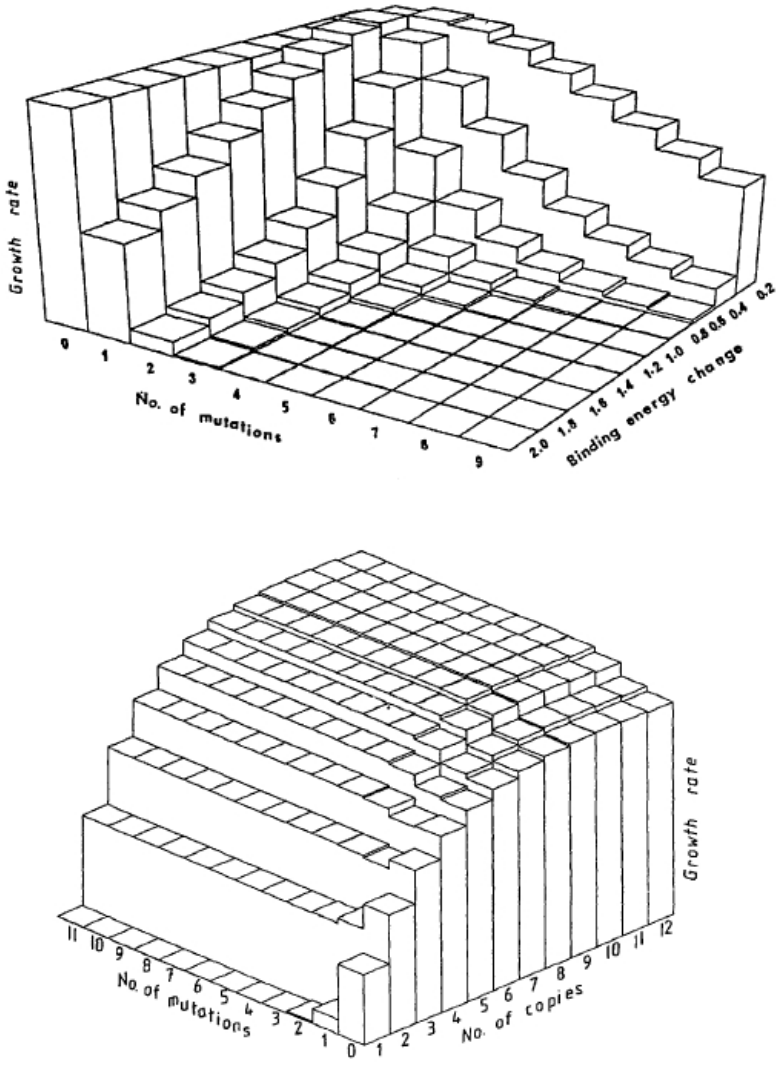


Figure 6: Enzyme evolution: Top = Mutational trap, Bottom = Escape from the trap [KB84].

Most of the models that are concerned with the question, discussed above, stem from reconstructing metabolic networks of a number of organisms from pathway information available in

databases like BioCyc[1] or KEGG[2]. They differ mainly in what properties of metabolic systems they integrate. It would be interesting to study these different hypothesis in a simulation environment.

2.3 Representation of Chemical Reactions

Chemistry basically consists of two parts: firstly, chemical structures or molecules which could be seen as the static part and, on the other side, chemical reactions which would rather represent the dynamic part of chemistry. A chemical reaction can be seen as the transformation of a starting structure into a product structure, where both structures may be several molecules. In this thesis we will only regard structures consisting of at most two molecules. In chemistry reactions are usually depicted as in figure 7, with the starting molecules on the left side and the product molecules on the right side of the arrow indicating the reaction direction.

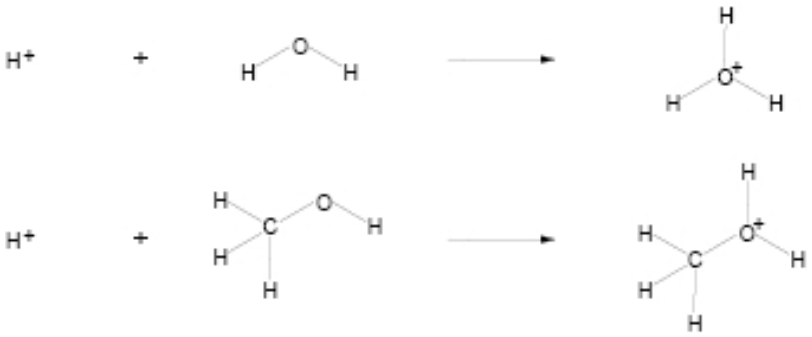


Figure 7: Typical representation of a chemical reaction

As one can imagine there is an enormous number of possible chemical reactions. Many millions haven been documented in literature, and there exist reaction databases which povide many of them (see table 1[3]) and enable the chemist to search for reactions containing certain properties, in the form of substructures. Unfortunately, because of the number of reactions in these databases the results of searches often, still, tend to be too copious to accurately analyze them in the traditional way. Nevertheless, the use of such databases and search tools will be indispensable in chemistry in the future. Thus, some efforts are made to find better solutions. The basic

[1] http://www.biocyc.org/
[2] http://www.genome.jp/kegg/
[3] http://www.mdli.com/

idea of all approaches is to classify reactions and create a certain hierarchy, with some coarse classifications on the top and the reactions as the leaves.

Databases	Number of reactions	Entries
CrossFire Beilstein	>10.000.000	Collection
ChemInform	≈1.200.000	Reaction Types
RefLib	120.000	Reactions
Derwent Journal s M	≈80.000	Reactions
Orgsyn	>5.000	Experiments

Table 1: Chemical reactions available in databases

2.3.1 Reaction Classification Systems

A reaction classification system tries to group the enormous number of chemical reactions into a few classes with a certain chemical meaning and further levels of subclasses which then indicate increasingly specific functional properties. The last level of this hierarchy, the leaves, would be the reactions themselves. There are two types of reaction classification systems: first of all, ”model-driven” classification systems which try to define a classification for all possible reactions using a model of the reaction center of a so called unit reaction; and second, there are ”data-driven” classifications which aim to classify a given set of reactions by analyzing their chemical properties, e.g. functional groups, in order to find common characteristics. Here we will only discuss the former type of reaction classification systems.

The model on which the ”model-driven” reaction classification systems base their classification, is, as mentioned, the reaction center of unit reactions. In particular, the way the bonds behave in the reaction center. A unit reaction is basically just a very simple reaction in which all involved atoms only redistribute a pair of electrons with neighboring atoms of the reaction center. The reaction center of chemical reactions contain only those atoms which are involved in bond change. The presentation of such a reaction is depicted in figure 8.

In particular, we are interested in classifications which use a representation of reactions comprising only one combining graph. Some of these approaches are depicted in figure 9. We will discuss the latter two in the following sections. The classification of Vladutz[F.D86] is very similar to and actually predates that of Fujita[Fuj86a], but latter is preferred since it provides a complete classification system and thus provides a better explanation. One approach -Arens’[Are79]- which stands out from the others, will be mentioned and partly described in the sections Enumeration (2.3.4) and Nomenclature (2.3.5) because it happens to be similar to

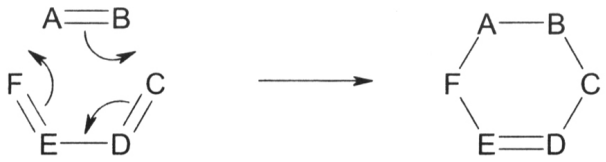


Figure 8: Unit reaction on the example of nitroso nycloaddition [Gas03].

the idea of indexing reaction introduced in this thesis, and consequently also used within the simulation tool.

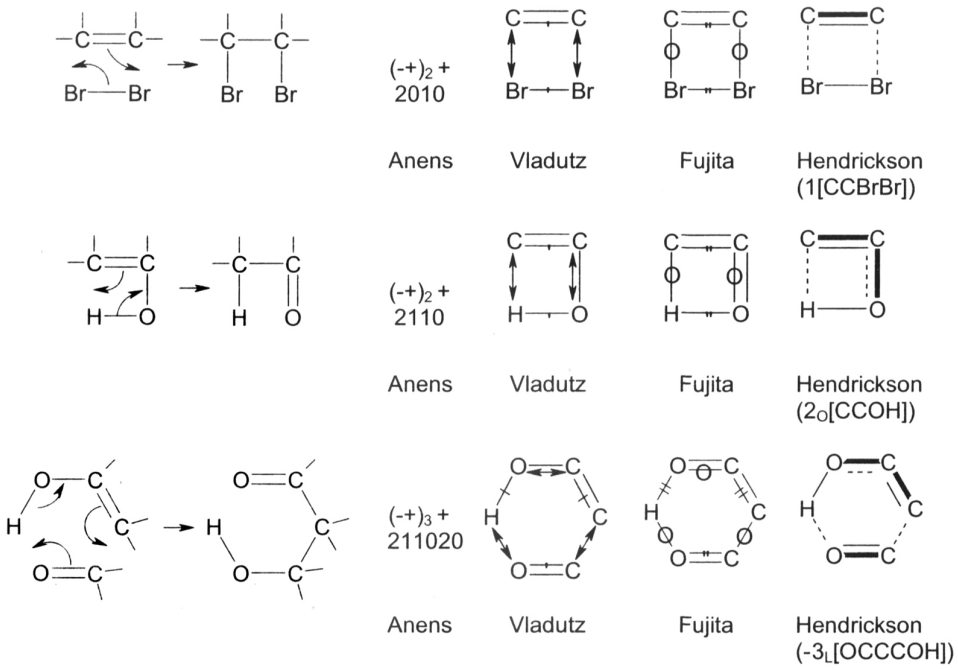


Figure 9: Different reaction representations [Gas03]

2.3.2 ITS - Imaginary Transition Structures

Both Vladutz and Fujita used so called Imaginary Transition Structures (ITS)[Fuj86a, Fuj87c], unitary graph presentations of chemical reactions, in their classification systems, but throughout the thesis it will always be referred only to Fujita's work since he provided an extensive catalog of classified reactions [Fuj86b, Fuj86c, Fuj87a, Fuj87b] and further going research based on his Reaction Center Graph (RCG)[Fuj86d, Fuj86e], such as nomenclature and enumeration. Thus,

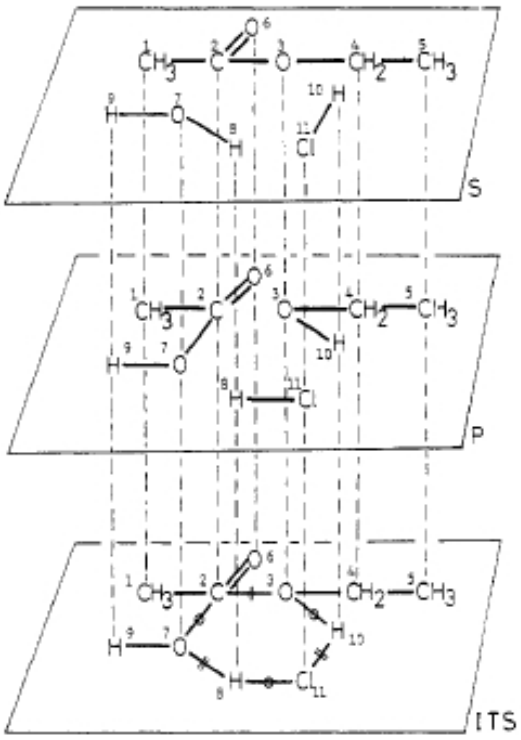


Figure 10: Superimposition of the ITS [Fuj86a]

it would be preferable to study his account if one wants to get more deeply involved in the matter of chemical reactions and classification systems.

An ITS is a graph similar to the representation of chemical structures, containing information about both the starting as well as the product molecule and superimposing the graphs of the two. Consequently, such a graph has the need for additional bond types. Since starting and product molecule enclose the same atoms, they are also contained in the ITS. For the graph edges it is differently. Besides the regular bond of a chemical structure indicating that two atoms are connected, there are two more bond types for the ITS. One bond representing connections which are only present in the starting molecule and accordingly another bond indicating connections contained only in the product molecule. If two atoms are connected in both graphs, the regular bond type is used. Figure 10 illustrates the superimposing of the two graphs to the resulting ITS.

The use of ITS would still lead to millions of different reactions and does not classify similar reactions into types. Thus, Reaction Center Graphs (RCG) are introduced by Fujita. These are

subgraphs of an ITS only containing atoms which are actually involved in the reaction. This is realized by allowing only atoms that are connected through the two newly introduced bond types for the RCG and discarding all atoms that have only connections contained in both graphs. These Reaction Center Graphs now can be used to describe different reaction types. Classes of RCGs can be distinguished by their stringity, i.e. by how many strings a reaction center can be described. Such a reaction string represents one route through the RCG with no duplicates and alternating between the two bond types mentioned above. In this thesis only reactions which can be represented by one string, thus have stringity of one, are considered, but theoretically reactions with many more strings can exist. In Figure 11 several reaction center graphs of one-string and two-string reactions are depicted.

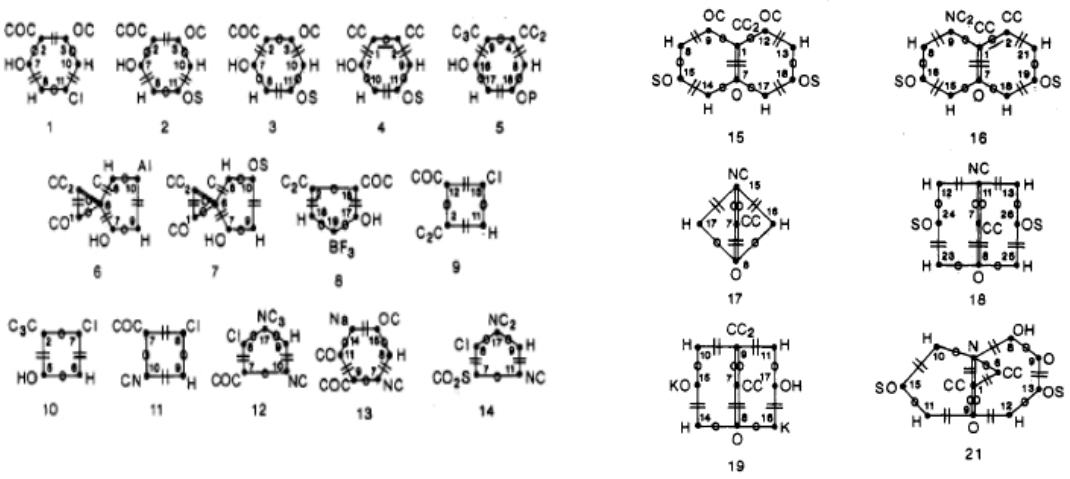


Figure 11: RCGs of one-string and two-string reactions [Fuj86a]

One-string reactions can be seen as rings in ITS graphs. Along this ring, bonds which belong only to the starting molecule and bonds which belong only to the product molecule alternate. Optionally, bonds being contained in both molecules can exist. The characteristics and the order of the edges of these rings are used by Fujita to classify reactions into certain chemical types. The one-string reaction center graph with all its edges is a closed ring. However, if one regards only those edges which are contained in both molecule graphs and those contained only in the starting molecule, then rings which are not connected at one or more points can occur. To be precise, there can be up to $n/2$ gaps in the ring, with n being the number of atoms. The same is true for rings containing the edges of the product molecule and those being present in both stages. We will call the number of gaps in the former ring s and for the latter p. If $s = 0$ and $p > 0$ then the respective reaction is considered to belong to the group of ring opener (RO) and

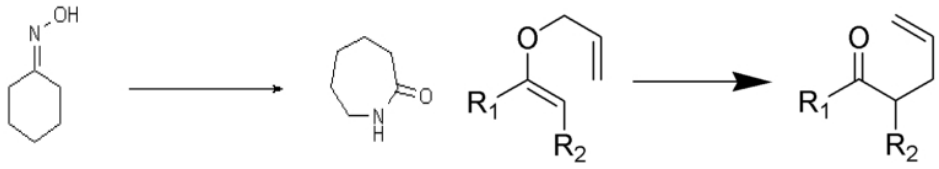


Figure 12: Beckmann and Claisen rearrangement

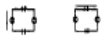
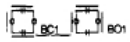
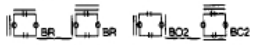
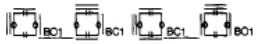
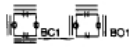
can be further classified based on p (e.g. RO2 if $p = 2$). A similar classification can be done with reactions where $p = 0$ and $s > 0$, which would be grouped, accordingly, as ring closer (RC) with s specifying the group further (e.g. RC1 if $s = 1$). Another group of reactions is called bridges of rearrangements (BR). Reactions falling into this group are, for example, the Claisen rearrangement or the Beckmann rearrangement (see figure 12). All reactions in this group have $s = 1$ and $p = 1$; thus, both, the ring in the starting molecule and the product molecule are not closed, but the position of the gap changes through the rearrangement.

Fujita provides drawings and classifications of all possible arrangements of bonds in rings from size three up to six, see table 2 for all possible reaction center graphs for one-string reactions with four atoms. In this thesis, as default, reactions ranging from three to six atoms are considered. If desired, this set can be extended to reactions with more atoms in the reaction center. Fujita also determined the number of possible RCGs for reactions of the different sizes with the help of Polya's polynomials, as will be discussed in the Enumeration section (2.3.4).

2.3.3 Hendrickson

So far reactions were only classified by the order and orientation of their bonds. Hendrickson also provides a classification regarding atom types in all possible positions[Hen74, Hen79]. The atoms of the reaction center are grouped based on their valency for classification purposes. Hendrickson's considerations are regarding reactions containing C, N, O and H atoms, the same set of atoms as will be used within this thesis. In his work, the different conformations and the number of possible reactions are given for reaction centers with five and six atoms. Another restriction is that at least two of them are C atoms. A similar constraint will be applied on our set of reactions as well.

In another work, Hendrickson presents a general representation for reaction types[Hen97], as introduced before by Fujita and others. In his reaction center graphs, atoms are aligned clockwise and ordered starting from the top left with the atoms of highest valence. Furthermore, the

m	n	no. of reacn graphs[a]	no. of reacn pairs[a]	reaction graphs
0	0	1	1	
0	1	2	1	
0	2	3	2	
0	3	2	1	
0	4	1	1	
1	0	2	1	
1	1	4	2	
1	2	4	2	
1	3	2	1	
2	0	3	2	
2	1	4	2	
2	2	3	2	
3	0	2	1	
3	1	2	1	
4	0	1	1	

[a] The numbers of reaction graphs are the coefficients of $x^m y^n$ in $G(x,y)$ (eq 7). [b] The numbers of reaction pairs are the coefficients of $x^m y^n$ in $P(x,y)$

Table 2: Reaction graphs of the tetragonal class [Fuj86b]

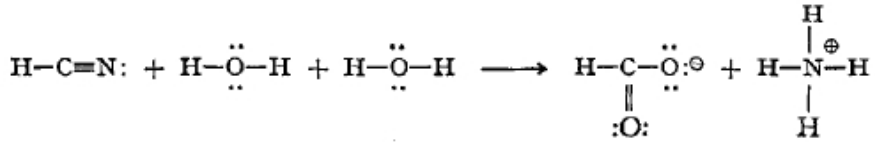
first edge within this order is always a breaking bond, i.e. a bond which exists in the starting molecule but not in the product molecule. These conventions allow for a unified representation and an easy way to a general nomenclature for reactions, as will be discussed in the section Nomenclature (2.3.5).

2.3.4 Enumeration

The discovery of new chemical reactions is of interest in chemistry and also poses an interesting task for computational chemistry. Several different approaches for the search of novel reaction types do exist. One of the earliest approaches stems from Ugi and Dugundji. Their reaction generation program -IGOR[BU88]- works with matrices[DU73] representing the chemical structures as well as the transition states, see figure 13. The program is interactive as the user has to enter the reaction matrices by himself. [Her90], on the other hand, describes a graph-based approach: by generating all non-redundant spanning subgraphs of a given graph, new chemical reactions are predicted. The algorithm works in a similar way as Fujita manually procedure to define his classification. For all approaches, the important part is to find all basic reactions to a certain reaction matrix or graph. Figure 14 illustrates the task for the example of six-cycle reactions.

Zefirov introduces a formal-logic approach to attack the problem and realizes this idea in a computer program - SYMBEQ[ZBP94]- which searches for new reactions. The underlying algorithm basically consists of two major steps, as does the procedure in this thesis. The first step is similar to the algorithm of Herges[Her90]. The graph from which the non-redundant set of spanning subgraphs is generated is here called topology identifier and the resulting subgraphs are called symbolic equations. The second step further exploits all possible reaction centers regarding the involved atoms. In Figure 15 the hierarchy of this algorithm as well as examples for the two steps are depicted.

In this thesis it is not intended to integrate all possible chemical reactions but instead complete sets for certain topology identifiers or reaction graphs. In particular, pericyclic reactions of size three to six. As Fujita explained in his classification work, the number of all possible reaction types for this class of reactions can easily be obtained applying Polya's polynomials on the number of involved atoms (see equations 1, 2 and 3). Equation 1 is the cycle index for six-cycle reactions. Equation 2 is the figure counting series that is substituted which leads to the Polya polynomial series of equation 3 for six-cycle reaction. The coefficients of $x^m y^n$ represent the number of reaction center graphs. The process of the enumeration can be best explained using a terminology from Herges and distinguish between to parts of the reaction graphs: one being



1 *2a* *2b* *4* *5*

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} \mathrm{H} \\ \mathrm{C} \\ \mathrm{N} \\ \mathrm{H} \\ \mathrm{O} \\ \mathrm{H} \\ \mathrm{H} \\ \mathrm{O} \\ \mathrm{H} \end{matrix}$$

$$\begin{matrix} \mathrm{H} & \mathrm{C} & \mathrm{N} & \mathrm{H} & \mathrm{O} & \mathrm{H} & \mathrm{H} & \mathrm{O} & \mathrm{H} \end{matrix}$$

$$E = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \mathrm{H} \\ \mathrm{C} \\ \mathrm{N} \\ \mathrm{H} \\ \mathrm{O} \\ \mathrm{H} \\ \mathrm{H} \\ \mathrm{O} \\ \mathrm{H} \end{matrix}$$

$$\begin{matrix} \mathrm{H} & \mathrm{C} & \mathrm{N} & \mathrm{H} & \mathrm{O} & \mathrm{H} & \mathrm{H} & \mathrm{O} & \mathrm{H} \end{matrix}$$

The reaction matrix of $1+2a+2b \rightarrow 4+5$ is $R=E-B$

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 0 & +2 & 0 & 0 & +1 & 0 \\ 0 & -3 & -2 & +1 & 0 & +1 & +1 & 0 & +1 \\ 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & +2 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & +1 & 0 & 0 & 0 & 0 & -1 & +2 & -1 \\ 0 & 0 & +1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{matrix} \mathrm{H} \\ \mathrm{C} \\ \mathrm{N} \\ \mathrm{H} \\ \mathrm{O} \\ \mathrm{H} \\ \mathrm{H} \\ \mathrm{O} \\ \mathrm{H} \end{matrix}$$

$$\begin{matrix} \mathrm{H} & \mathrm{C} & \mathrm{N} & \mathrm{H} & \mathrm{O} & \mathrm{H} & \mathrm{H} & \mathrm{O} & \mathrm{H} \end{matrix}$$

Figure 13: Matrix representations of Ugi-Djugundji [DU73]. B = substrate, E = product, R = Reaction.

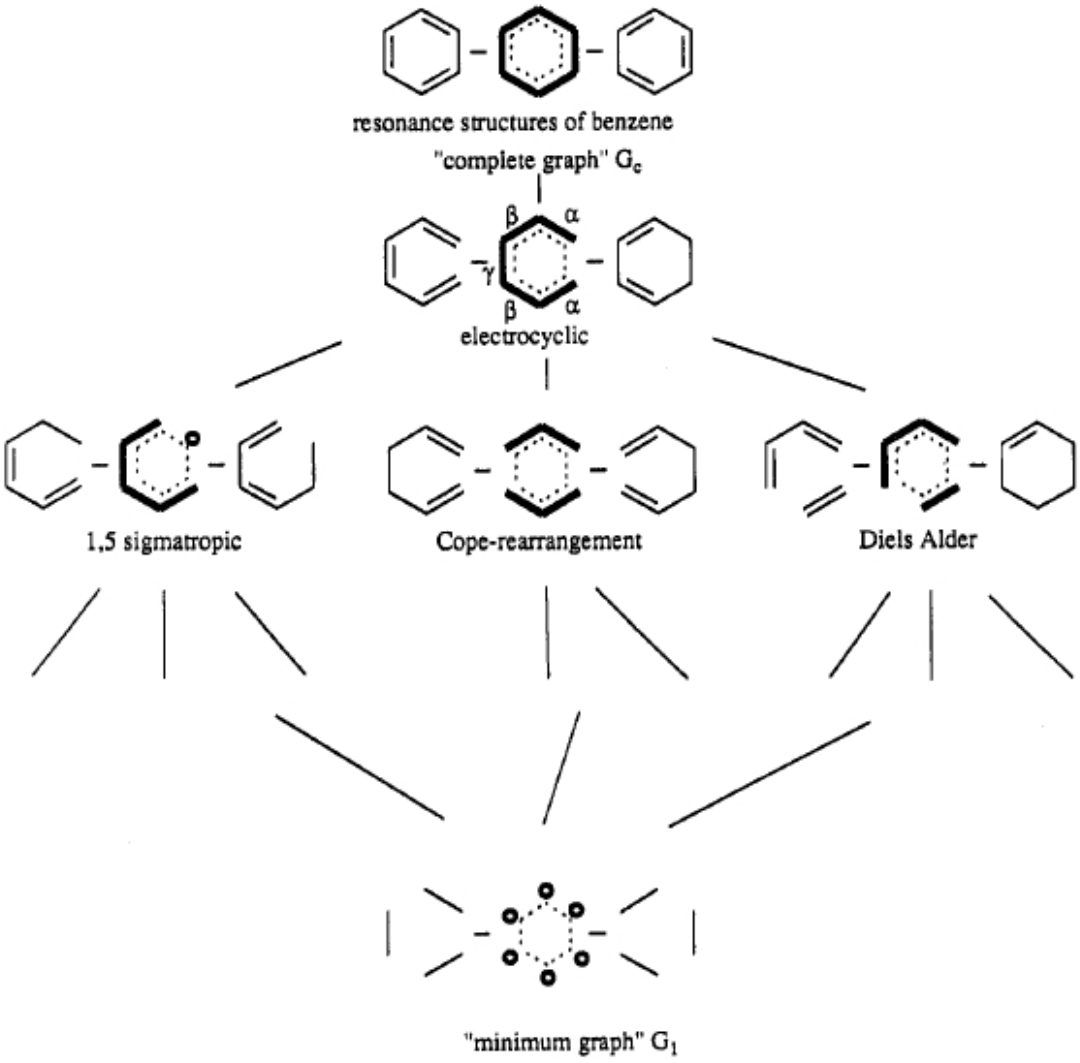


Figure 14: Enumeration of all basic reactions for 6-cycle reactions [Her90]

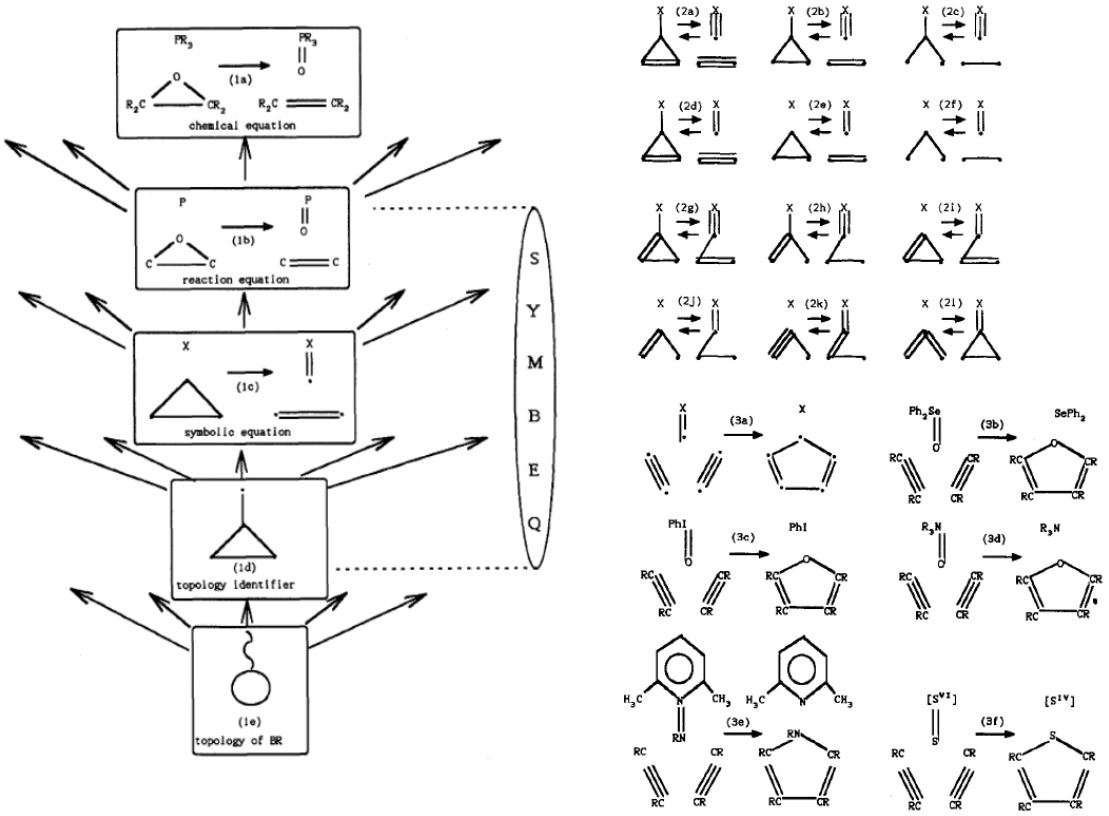


Figure 15: SYMBEQ: Left = Hierarchy. Right: Top = Step 1, Bottom = Step 2. Step 1 = Enumeration of all possible symbolic equations from the topology identifier 2a. Step 2 = Enumeration of all reaction equations from symbolic equation 3a. [ZBP94]

the bond changes which occur throughout the reaction, and the other being something like the σ-frame of the reaction. The former type of edges is the same for all the generated graphs for one reaction class, while the latter can have different structure, from the empty frame to a full frame containing either single or even double bonds, determining the particular reaction type. It is now easy to enumerate all of these graphs, but obviously the resulting set would contain many duplicates. Consequently, we do have to check each graph for redundancy. Since it is desired to avoid to do subgraph-isomorphism checking for all graphs against the entire set, an order among the graphs will be generated and only minimal graphs will be allowed into the final set. Since the size of this set is known through the calculation of Polya's polynomials, the algorithm can terminate without constructing all possible graphs.

$$Z(D_3) = (1/6)(s_1^6 + 3s_1^2 s_2^2 + 2s_3^2) \tag{1}$$

$$s_k = 1 + x^k + y^k \tag{2}$$

$$\begin{aligned} G(x,y) &= Z(D_3, 1 + x^k + y^k) \\ &= 1 + 2x + 2y + 4x^2 + 6xy + 4y^2 \\ &\quad +6x^3 + 12x^2y + 12xy^2 + 6y^3 + 4x^4 \\ &\quad +12x^3y + 18x^2y^2 + 12xy^3 + 4y^4 \\ &\quad +2x^5 + 6x^4y + 12x^3y^2 + 12x^2y^3 \\ &\quad +6xy^4 + 2y^5 + x^6 + 2x^5y + 4x^4y^2 \\ &\quad +6x^3y^3 + 4x^2y^4 + 2xy^5 + y^6 \end{aligned} \tag{3}$$

In terms of the hierarchy used in SYMBEQ (figure 15), the first step is completed and the entire set of non-redundant symbolic equations or in our case reaction graphs, according to Fujita, is derived. In SYMBEQ, the subsequent step leads from symbolic equations to the set of reaction equations. Accordingly, in our approach, atom types are assigned to the nodes of the reaction graphs. Obviously, there exist numerous different arrangements of atom types for each reaction graph. In this work, only a small set of atom types (H, O, N, C) will be used, and restrictions like the minimal number of C atoms in a reaction center will be applied. First of all, it is believed that many important reactions can still be generated and, secondly, it constitutes a similar notation as Hendrickson in his examples by using atom types for the valences from one (H) to four (C). For the enumeration of atom type arrangement, the same problem of redundancy emerges as in step one. Correspondingly, a similar approach is used here to solve it. Again we will define an order among the resulting graphs and only add minimal graphs to the final set. Important for the orders in both steps is that the bond changes are fixed, e.g. the first edge always is a breaking

bond. In step one the order was among the edges, e.g. double $>$ single $>$ no bond, whereas in the second step the order is among the atom types, e.g. H $>$ O $>$ N $>$ C.

2.3.5 Nomenclature

Another task desired to be solved is the definition of a general nomenclature for reactions, e.g. a reaction string or reaction index. Two interesting approaches for this problem are those of Arens[Are79] and Hendrickson[Hen97]. The index used in this thesis is inspired by the ideas of both of them. Although all nomenclatures may differ more or less in their structure, they all have to include certain very similar conventions. Those conventions are regarding the bond order. One degree of variability is reduced by restricting the first edge to be a breaking bond, i.e. a bond that is contained in the starting molecule but not anymore in the product molecule. The next convention concerns the edges contained in both molecule graphs. This convention is realized differently in the two approaches. Hendrickson uses certain symbols to describe the arrangement of the bonds in the reaction center graphs, which were introduced by Balaban. The symbols can be seen in the column shell bonds in Figure 16. Although this notation seems more compact in the example, it is here only shown for the cases of single bonds in the sigma-frame and does not consider double bonds. Usage of double bonds will make the entire notation much more complicated. Hence, we will concentrate more on Arens' solution, which uses an order among the bonds. This is very practical since we already introduced an order among those bonds for the problem of enumerating reactions. Arens applies his order upon the bonds of the starting molecule and then lies a fixed pattern of the changing bonds on top of it. In this thesis, however, the order will be applied onto bonds of the reaction center and the bond-change pattern will be implied through the convention made in the beginning. The advantage of the index introduced in this thesis over that of Arens is that it resembles the same order of atoms and bonds as if one would read it from the reaction center graphs of Fujita or Hendrickson starting on the top left corner. Furthermore, Arens does not include the atom types in his nomenclature, however, this is a rather easily solved problem. Similar as Hendrickson, one could write the atom types as one sequence behind the information for the bonds. In our index, the atom types are written such that the information about the respective edge is between the connected atoms. Also, the valency of the atom will be used as information for the atom type, since only the reduced set of atom types with a unique relation between valency and atom type is regarded (H=1, O=2, N=3, C=4).

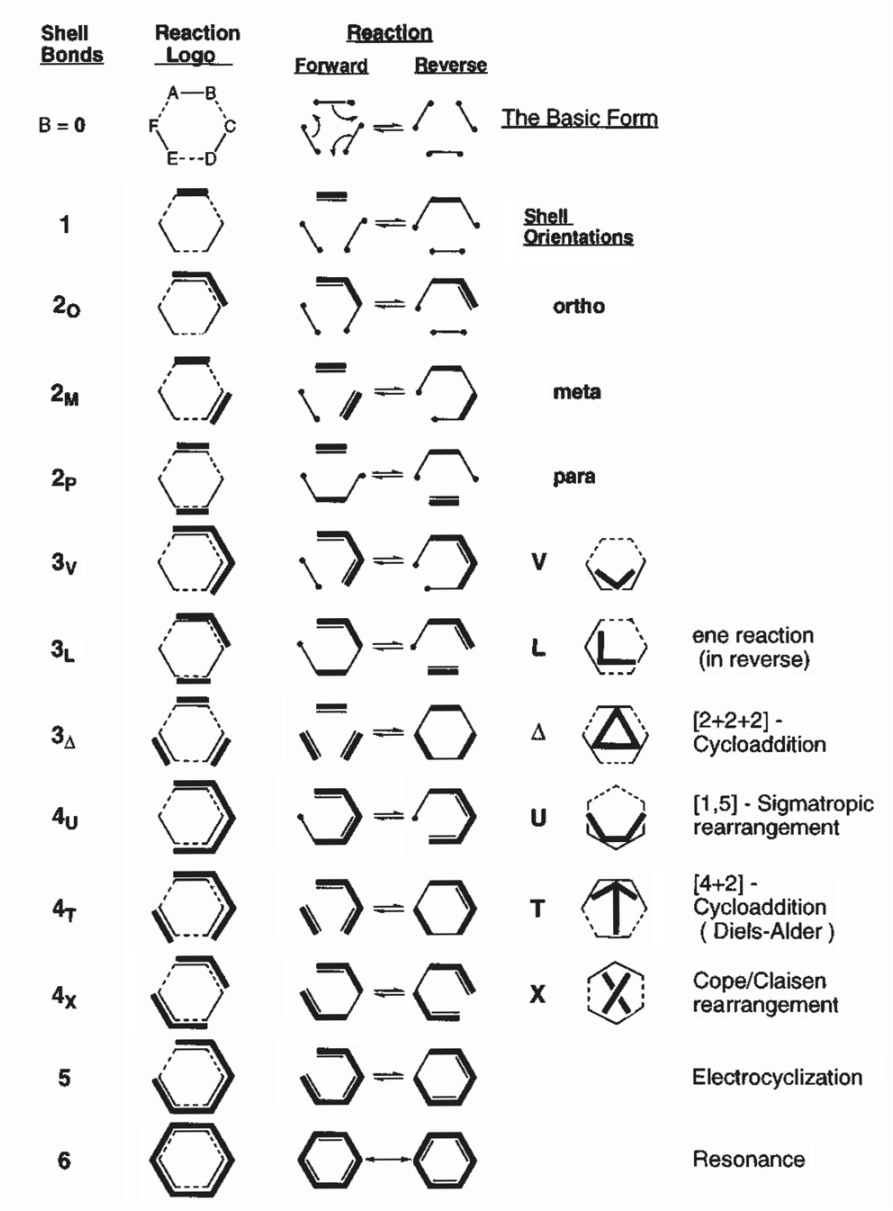


Figure 16: Reaction logos for 6-cycle reactions, grouped by shell bonds [Hen97]

3 The Model

In this chapter the basic framework of the simulation, the model, will be discussed. Starting with the big picture to illustrate the idea and give an overview of the participating components and their connections, leading in later sections to a more detailed description of the different parts and their roles in the model as well as in the implementation. The intention here is to give an understanding of how abstract things as evolution or complex things as metabolism are realized in this computational approach.

3.1 Overview

Before the individual components are discussed, the process as a whole will be explained. Figure 17 shows all of the components involved in the simulation and and the way by which they are related to each other. Components in squares are structures or objects, i.e. they contain certain objects and have computational representations like sets or vectors. The remaining parts which are indicated through a circle or ellipse are processes, i.e. they are actions or, computationally speaking, algorithms applied on other components which are objects. Arrows indicate the direction of influence between the different parts. And, finally, there is a distinction between user defined parts, those situated left of the dashed line, and the dependent parts on the right side of the line.

As can be seen, there is one superior concept which contains all interior objects. This concept is called `Individual` and its instances represent single cells or metabolisms. Through this concept the process of Evolution is modeled and realized. We start of with a population of a certain number of individuals, i.e. instances of the concept Individual, and let them build up a metabolism based on their set of enzymes out of the entire set of chemical reactions and the metabolites of the environment which were chosen initially. In the following generations the metabolism of a cell is generated from the already existing metabolism, the set of chemical reactions and the metabolites from the environment together with those that were build in previous generations. The metabolism is represented by a network graph -the `Metabolic Network`- from which certain network properties can be derived in order to compare the different metabolisms and thus, the corresponding individuals. The comparison is realized by metabolic flux analysis which is explained in detail in the corresponding chapter (5). Having a measure for the individuals, a set of optimal individuals can now be selected in each generation. From these individuals new individuals are generated so that the population size is kept steady. The newly generated individuals contain the same metabolic network and metabolite pool as their parents in the beginning, but do have a mutation in their genome which

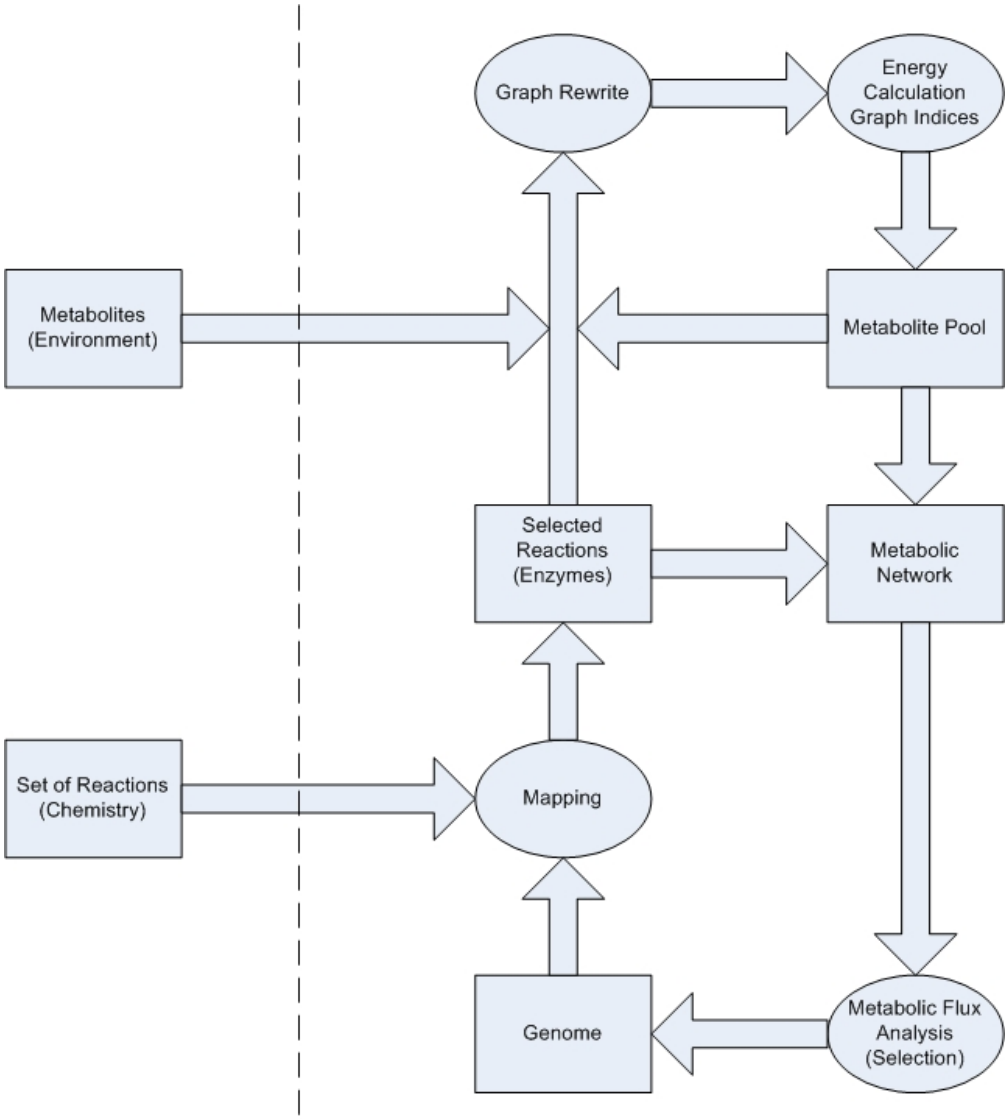


Figure 17: Overview of the Model

might result in a different set of enzymes, i.e. chemical reactions. Consequently, the resulting metabolic network may be different.

3.2 Individual

As mentioned above, the concept of Individual is superior to the other components of the model, i.e. it contains all of the objects which we termed dependent or internal objects. Each instance of Individual contains a genome, a metabolite pool, a set of enzymes and a metabolic network. Thus, we are able to treat it as a representation of a cell with metabolism. Furthermore, individuals can be connected with each other, e.g. son-of, father-of, to indicate that a certain individual was generated by another. This allows us to analyze the phylogeny of a population after the simulation and compare them to populations from other simulation runs initialized with different environments or settings in general. These connections are realized by carrying a personal historic number and the historic number of its parent.

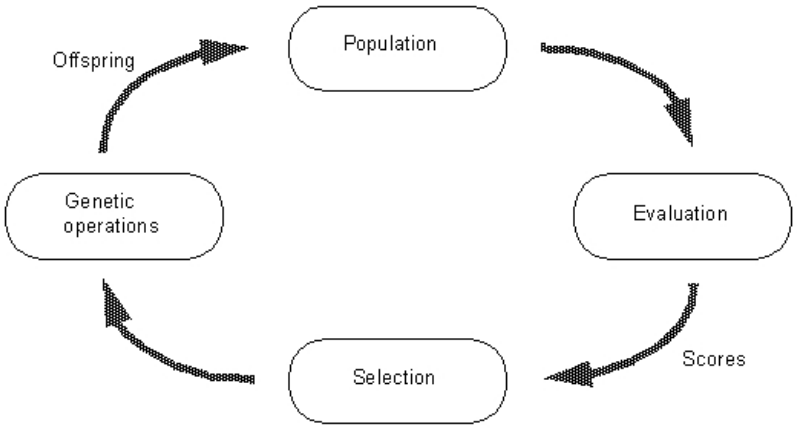


Figure 18: Process of evolution in the model

Internally, the population is realized by a simple vector of individuals. The population size is one of the parameters set initially. Also the number of individuals which ought to be selected in each generation has to be specified and from this follows the number of offspring.

3.3 Genome

Every individual contains a genome of a fixed length and a common TATA-box sequence. Furthermore, all genes have the same length. The genome is a RNA-sequence and the single genes are supposed to represent RNA-enzymes and bear the function of a particular chemical reaction

out of the entire set of reactions derived through the ITS-enumeration. More detailed description on that is provided in section Reactions (3.5) in this chapter and in section Representation of chemical Reactions (2.3). In each generation, new individuals are generated from the set of optimal individuals. Those new individuals contain a copy of the parent genome to which a point mutation was applied. The mutation can occur everywhere in the genome; thus, there can be silent mutations, where the mutation takes place in a non-coding region, or neutral mutations which change a nucleotide within a gene but not the function of the corresponding RNA-Enzyme, i.e. it still performs the same chemical reaction. Of course, there also can be missense mutations which change the structure of the RNA-Enzyme in such a way that it inhabits a different function than before, and there can be mutations which either destroy a TATA-box (nonsense mutation) or build a new TATA-box and thus eliminate or add a new gene to the genome, respectively.

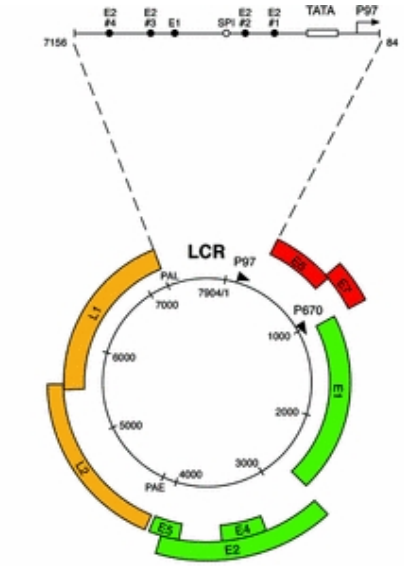


Figure 19: Abstraction of the genome in this model

The genome is realized by a string containing the nucleotide sequence and a list of all genes which have to be transcribed to RNA-Enzymes. The sequence is treated as circular; thus, there are as many genes as there are TATA-boxes and some of the genes may reach over the ends or overlap. In the course of the enumeration of the chemical reactions (see section 2.3.4), an ID was assigned to each reaction. The ID's for the currently expressed reaction and the reaction of the parent gene are stored in each gene and all the genes, active or not, are listed in the genome. With this information we can retrace the history of every single gene and determine whether it

had a single or multiple origin and whether it may have disappeared for some generations before reappearing. Furthermore, the entire history of mutations is kept in the genome. Hence, we are able to determine the exact time of change and analyze the means of these changes. This also allows us to compare sequential and structural changes with the external events, e.g. drifting environment, selection pressure.

3.4 Metabolites

Every individual contains a list or as we will call it here a pool of metabolites. In the beginning, this pool consists only of the metabolites of the environment and is then extended by the products of its metabolism, i.e. the application of all its enzymes or chemical reactions. The environment is chosen initially and can even be made to change in the course of evolution. To avoid redundancy in the computation of products, reactions which did not underlie change are only applied to metabolites which were generated in the last generation. Metabolites have to fulfill a few restrictions in order to be generated by the reaction rules. One mandatory restriction in this approach is the energy value of a metabolism, it may not exceed a certain threshold and also the energy value of a product has to be lower or equal to that of the educt. For more detailed description see section Energy Calculation (4.3.2) in the next chapter. There is also an option to force metabolites to fulfill certain properties based on graph indices. Information about the different indices can be referred from section Topological Indices (4.4) in the next chapter. With these restrictions, the development of the metabolites can be guided in a more realistic direction. On the other hand, observing restrictions which show no significant difference in their metabolite pool than the original, may indicate principles of self-organization or at least suggest an evolutionary necessity, obviously, the restriction also just may not be very expressive or meaningful.

Metabolites are mainly represented by labeled graphs, but sometimes also occur as a unique structural formula in the form of a string or completely abstract as an ID. For the application of the chemical reaction, which are represented as a graph, we need the metabolites as graphs. The function of an enzyme then comes down to graph operations, e.g. joining and splitting of graphs, subgraph isomorphism, adding, removing and exchanging of edges (see chapter Graph-Based Toy-Universe). For the output of the metabolite pool, a unique and expressive structural formula is preferable because we want to protocol a large number of metabolites, here we will use `SMILES` due to its good interpretability. Another use of the `SMILES` representation becomes apparent when checking a metabolite for redundancy against the entire metabolite pool. It is much faster to compare two strings for equality than to perform a graph isomorphism check. The computation of the SMILES formula is rather negligible compared to the repeated checking for redundancy. For the use in the network graph, it suffices even to represent the metabolites

as an ID which is unique within this particular individual. The advantage lies in the clearer illustration when using a shorter description.

3.5 Reactions

One of the first steps is the enumeration of all reactions that are supposed to be included in the chemistry of the simulation. For different purposes one might want to choose different sets of chemical reactions, e.g. sometimes only reactions working on carbon-skeletons are of interest or in another experiment only reactions involving a small number of atoms are to be observed. Due to the sheer endless number of possible chemical reactions, the set of reactions which can be chosen in the simulation is restricted here to those containing hydrogen, oxygen, carbon or nitrogen atoms. We believe that these atoms suffice to build up the most important molecules necessary for a primitive metabolism as one would expect in the early evolution of metabolism and that the simulation still resembles a realistic account for the processes at such a phase or comparable situations and does not sustain a loss of expressiveness regarding robustness and other network properties. Furthermore, a reaction may only involve three to six atoms and not more than two metabolites. If we say that a certain number of atoms is involved in a reaction, then this does not mean that the metabolite which is to be worked on contains only this particular number of atoms, but rather that only the connections within a set of atoms of certain size is changed by the reaction. Figure 20 shows examples of reactions for different sizes of these sets. Most of the already known chemical reactions lie in this range and it can be assumed that, accordingly, reactions crucial to simple metabolisms or the most basic pathways and networks underlying all metabolisms can be found there. Reactions involving more atoms or metabolites, account only for few interesting reactions and would simply add to the complexity of the computation.

All reactions are internally represented by a labeled graph and a unique ID. Similar to the graphs for metabolites, both vertices and edges are labeled and represent atoms and bonds, respectively. The main difference between the two graph types is that reaction graphs have two labels for the edges. This is because they actually represent two graphs, one for the educt and one for the product of the respective reaction. The first one is used as the pattern in the subgraph isomorphism check against all metabolites and the second one will replace the found subgraph in the metabolite graph matching the pattern. For the application of so called bimolecular reactions, combinations of pairs of metabolites have to be checked for subgraph isomorphism. Thus, two metabolite graphs have to be joined before. Since these reactions basically contain two patterns, the combinations can be reduced by first checking all metabolites against the simpler patterns and then only joining pairs which contain each one of the two patterns. Both, bimolecular and

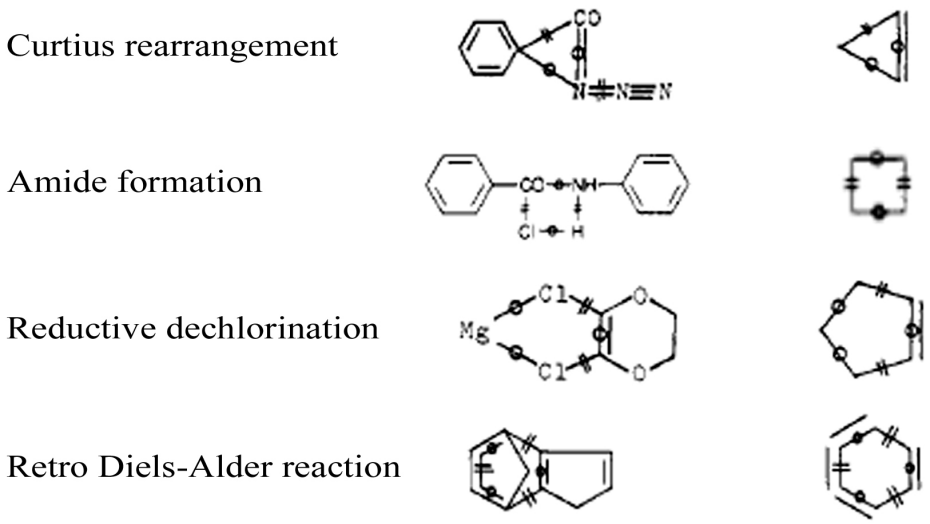


Figure 20: Reactions of different sizes

monomolecular reactions might result in two metabolites. In such a case, the metabolite graph has to be split into two graphs. More detailed information to the entire process can be found in the next chapter in section Graph-Rewriting (4.2).

We already mentioned that each gene is transcribed to a RNA enzyme and thus a reaction. This is possible because within the ITS enumeration process, ID's were assigned to all the reactions ,and genes expressing a certain reaction also bear the respective reaction ID. To know which gene belongs to which reaction, some properties of the gene transcript have to be mapped onto the properties of the reaction.

3.5.1 Mapping

Each gene is a RNA nucleotide sequence of fixed length and can, thus, like every other RNA sequence be folded into a secondary structure. This is here done by `RNAfold`[4]. The mapping does not consider the entire fold but rather one particular region, the longest loop within the fold, and adjacent stems of that loop. First of all, the length of this loop determines the number of atoms being involved in the reaction to which the gene will be mapped. A statistical analysis was performed when setting the mapping from loop length to x-cycle type, to ensure that the dif-

[4]http://www.tbi.univie.ac.at/ ivo/RNA/

ferent reaction types occur in appropriate proportions. The loop is then divided in as many parts as atoms are involved in the reaction. The mapping to the atom types of the reaction is derived simply from the sequence information in the different parts of the loop, each corresponding to one atom. The exact mapping from sequence information to atom type here is not important since not biologically meaningful. It suffices to notice that all atom types are chosen with the same rate. The bond type of the reaction logo (see section 2.3) is derived from the structural information of the different parts of the loop, in particular, the stems contained in these parts. The number of stems in a loop region, the length of these stems, and the sequence of the first two stem pairs accounts for the decision to which bond type will be mapped. Again the exact procedure of the mapping will not be discussed because it is a rather technical detail. However, one example is covered in figures 21,22 and table 3 for better understanding.

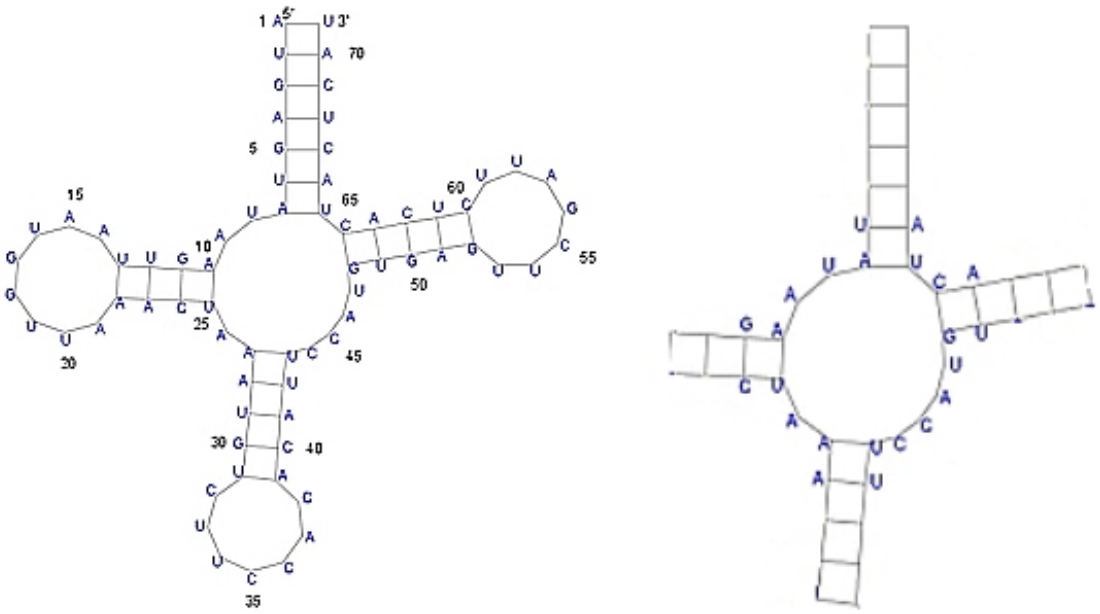


Figure 21: Example: Reaction mapping. Left: The folded RNA. Right: Longest loop of the folded RNA and used sequence information.

It has to be noted that the created RNA enzymes are not meant to be biologically meaningful in reality, they are just an abstraction of the idea that there were enzymatic molecules in the early stages of evolution that were RNA-based. Their function, similar to RNA enzymes known today, might have depended on the structure and the sequence of some few important regions of the entire molecule, as is suggested by the observation of catalytic sites in known RNA enzymes.

Section	Loop	C-G pair	Neighbor > 10	**Bond**	Valence	Seq. (loop)	**Sequence**
1	yes	0	yes (+1)	**1**	3	4	**4 = C**
2	yes	1	yes (+1)	**2**	4	1	**4 = C**
3	no	-	no	**0**	3	4	**4 = C**
4	yes	0	no	**0**	1	4	**4 = C**
5	no	-	yes (+1)	**1**	2	2	**2 = O**
6	yes	1	no	**1**	3	3	**3 = N**

Table 3: Example: Information derived from the longest Loop of the folded RNA-sequence

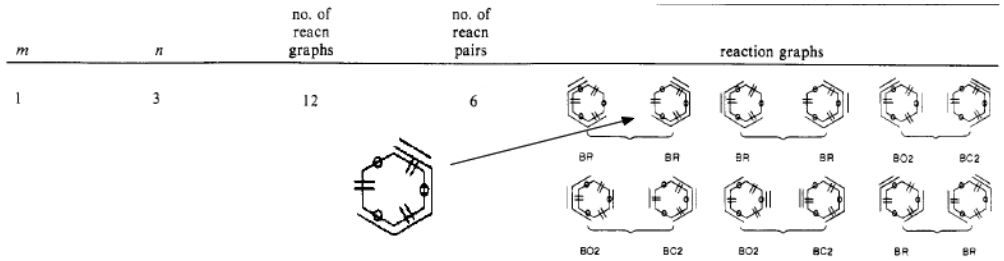


Figure 22: Example: Mapped reaction

For example, the function of the Hammerhead-RNA depends only on a few nucleotides at a certain position, see figure 23. The main reason for the rather technical details of the mapping are of statistical nature. First, it is important that all reactions are mapped to at approximately equal rates because it ought to be avoided to give some reactions an advantage in advance and thus bias the outcome of the resulting set of reactions. Secondly, and not less importantly, the mutation of genomes and genes should realistically affect the function of the enzymes. In reality, single point mutations can lead to missense, nonsense but also to silent mutations. When regarding RNA enzymes this makes much sense because a single change of one nucleotide usually does not change the fold of the transcript but may cause minor changes to the function if it occurred at the catalytic site. On the other hand, if the mutation destroys the original fold, it is also likely that the new enzyme will bear a completely new function. Some of these properties are already ensured by using the widely accredited `RNAfold` as the basis of our mapping. The properties of catalytic sites is accounted for in focusing on the longest loop and its structure, sequence and surrounding.

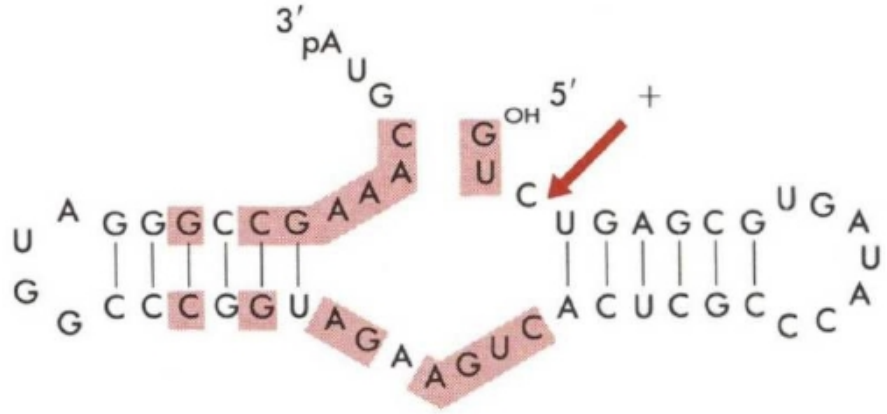


Figure 23: The hammerhead rybozyme

3.6 Metabolic Network

The central subject of the simulation is the metabolism, thus, we need a representation that we can easily observe and also use for analysis tools, in particular, here the metabolic flux analysis but other forms of network, graph or even grammar analysis as well. The most intuitive solution seems to be to use a network graph. In case of a metabolic network this could be a bidirectional and bipartite labeled graph. Bipartite because enzymes are only connected to metabolites and not to other enzymes and, vice versa, metabolites are only connected to enzymes. Bidirectional because one metabolite may at one time be the product of an enzyme and another time be the educt of the same enzyme, therefore, the direction of a connection is important. The nodes are labeled with ID's for metabolites and enzymes. The edge labels contain information about the specific reactions in which an enzyme-metabolite pair was involved. This is necessary because we can identify the exact parts of a reaction which can be up to four metabolites and one enzyme. Further, a metabolite can be the educt or product of an enzyme in more than one reaction. From this graph the stoichiometry matrix can easily be derived and it has the advantage that no information is lost and can be extracted in a straight forward way for almost all objectives. For example, the single reactions can be listed with educts and products. Consequently, it is possible to analyze from which different sources a metabolite can be gained. Looking at the enzyme graph may even enable to specify the exact regions which were joint, split or changed. The interpretability and expressiveness of this network graph, therefore, allows for a very detailed manually analysis as well as the typical computational approaches.

4 Graph-Based Toy-Universe

All structures in the simulation are modeled as graphs and processes are performed through applications on the structure graphs or the analysis of those. The choice to use graphs as the presentation for the structures in our chemical environment can be justified firstly by the fact that in chemistry molecules are for many years represented in graph form. Also it is the most intuitive way to regard chemical substances. Besides, networks are best understood by looking at its graph representation. As shown in the section Representation of Chemical Reactions (2.3), even for reactions and thus enzymes can graphs be used as appropriate models. Furthermore it is hoped and believed that using graphs for all parts of the model results in a more realistic behavior of the entire system as in other approaches. It should also be mentioned that there exist versatile applications which can be performed on graphs to analyze and transform them.

The graph-based model is supported by an artificial chemistry, `ToyChem`[BF05b], completing the universe in which individuals and their metabolisms can evolve. The artificial chemistry also works with a graph representation and promises to provide the look-and-feel of a real chemistry[BF03], certainly, integrating a chemical behavior sufficient for our purposes and more realistic than has so far been at the disposal of a comparable simulation approach[Ben02].

4.1 Graphs

In the model for the simulation, exist graphs for the metabolites, enzymes and the metabolic network. Furthermore, graphs in the form of phylogeny trees are generated for the purpose of protocoling and analyzing the evolutionary relationships of all genes as well as for individuals. The three first mentioned graphs are internally modeled as separate `C++` class implementing the Graph Interface of the `C++` meta-programming template library `boost`. For metabolites and enzymes, each exist alternative representations which are mainly needed as input format.

The metabolite graph is modeled in the `C++` class `metabolite.cpp` and inherits from the `Graph` interface, thus, having routines for the enumeration of vertices, adjacent vertices and edges. The vertices of the metabolite graph are the atoms of the respective chemical molecule and edges exist between vertices whose atoms are connected in the metabolite. The graph also contains several different labels. The vertices have labels for the atom type (H, O, N, C) and the edges use the bond type (-, =, -=) as label information. Further labeling is needed for the performance of the subgraph-isomorphism check. Labels for both vertices and edges exist, indicating whether they were already visited in the search for a matching subgraph or are already contained in the current subgraph mapping. There are a few situations in which metabolites

occur in another form: in the input file containing the metabolites constituting the environment, they are in sequential line form, SMILES[Wei88]. The same string format is used for the output of the current metabolite pool in the protocol and finally it is also used in the graph-isomorphism check performed after the graph-rewriting, as will be explained in the section Graph Rewriting (4.2).

The enzyme graph is modeled in the C++ class enzyme.cpp and also inherits from the Graph interface. Here the atoms and bonds of the reaction center of the corresponding reaction constitute the vertices and edges, respectively. Each vertex in an enzyme graph is connected to two other vertices in such a way that the atoms build a cycle. The vertex label is equal to that of the metabolite graph, but the edge-labeling differs somewhat. In the enzyme graph, every edge has two labels for bond-types: one for the substrate molecule and the other for the product molecule. Also the bond-types for enzymes are extended by the empty symbol, indicating that two atoms are not connected. During the subgraph-isomorphism check, labels for the vertices and edges contain information about the metabolite vertices and edges which can be mapped onto the substrate molecule part contained in the enzyme graph. There exist two alternative representations for enzymes: one is the GML format which is only used as manual input of reactions which ought to be used as the chemistry of the simulation; the other is a reaction-id, see section Nomenclature (2.3.5), which is used for the output of reactions in the protocol and also for the process of transcribing the RNA-enzymes from the genome of the respective individual. Further details can be found in the paragraph Mapping (3.5.1) of the preceding chapter.

The network graph is a direct instantiation of the Graph template because the integrated functions suffice for the needed purposes in the project. It differs from the other two graphs as it is a bi-directional graph, whereas, the graphs for metabolites and enzymes are undirected graphs. As already mentioned in the section Metabolic Network (3.6) in the previous chapter, the vertices of the network graph represent either metabolites or enzymes and they are labeled with id's uniquely identifying the respective substances for an individual. The edges of the network graph are somewhat different than that of the metabolite and enzyme graph. For one, the edges are directed and, furthermore, no two metabolites nor two enzymes can be connected; only a pair containing both types can be connected since an edge resembles an interaction on the molecular level. If an edge is directed from a metabolite to an enzyme, the metabolite has the role of the substrate molecule in the reaction, otherwise, it resembles the product molecule. In each chemical reaction, at least two and at most four metabolites (1-2 substrates + 1-2 products) and one enzyme are involved. Furthermore, a reaction id indicating the connection between them is assigned, which then serves as the edge label information. If an edge already exists, its label is

updated by adding the new reaction to the current label, separating them with a comma.

It should be noted here that the external `ToyChem` package also uses graph representations, but the details of those will not be explained in this thesis. Some information about the orbital graph and `ToyChem` in general will be provided in the section Energy Calculation (4.3.2). Important for us at this point is only that the energy calculation of `ToyChem` solely needs the `SMILES` format of the respective metabolite as input. The exact internal graph representations are of no direct interest for the simulation itself.

4.1.1 GML

For the input of the set of chemical reaction graphs, constituting the chemistry of the system, an appropriate format is needed. The `GML`, Graph Modelling Language[Him], is a flexible format with a simple syntax, meeting all requirements on a graph input format. `GML` is in `ASCII` representation, thus, ensuring portability regarding the platform and simplicity concerning the use of parsers. Furthermore, the structure of `GML` can be regarded simple. It consists basically of hierarchical key-value lists, where keys are alphanumeric characters, such as `graph` or `node`. Values can be integers, float numbers, strings, or another key-value list and can contain attributes. Further attributes and keys can be specified since `GML` intends to be implemented for many different data-structures.

For the modeling of graphs, there exist three different keys. The top level key -`graph`- containing the other two, `node` and `edge`. For the purposes of the simulation model, the syntax of `GML` was extended. Instead of `graph` the new key `rule` is used to define a reaction graph. The node `rule` always contains the keys `context`, `left` and `right`, all of which are newly added. In `GML`, graphs can contain keys `node` and `edge`. The topological structure is modeled through the `node id`'s used in the `edge`'s `source` and `target`. The reaction graphs are defined in the same way, but `node` keys are exclusively listed in `context`. Both, `left` and `right` contain `edge` keys. All `node`'s have the attribute `label` representing the atom type, whereas, `edge`'s are labeled with the bond type of the respective edge in the reaction graph. Below, the Diels-Alder reaction is shown in the changed `GML` format. The `context` specifies the atoms of the pericyclic reaction; `left` defines the edges of the graph resembling the substrate molecule; `right`, accordingly, defines the product molecule graph.

```
# ID 414141404140
rule [
  context [
    node [ id 0 label "C" ]
    node [ id 1 label "C" ]
    node [ id 2 label "C" ]
    node [ id 3 label "C" ]
    node [ id 4 label "C" ]
    node [ id 5 label "C" ]
  ]
  left [
    edge [ source 0 target 1 label "=" ]
    edge [ source 1 target 2 label "-" ]
    edge [ source 2 target 3 label "=" ]
    edge [ source 4 target 5 label "=" ]
  ]
  right [
    edge [ source 0 target 1 label "-" ]
    edge [ source 1 target 2 label "=" ]
    edge [ source 2 target 3 label "-" ]
    edge [ source 3 target 4 label "-" ]
    edge [ source 4 target 5 label "-" ]
    edge [ source 5 target 0 label "-" ]
  ]
]
```

4.1.2 SMILES

For the input and output of metabolites, a notation for chemical compounds is needed. SMILES, Simplified Molecular Input Line Entry System, provides a notation that is short and readable for chemists [Wei88]. Therefore, it accounts for an appropriate format for the output of a large set of metabolites, as is done in the simulation protocol. Despite the number of metabolites being contained in a metabolic network, it is still manageable for manual analysis through the SMILES representation. Another advantage of SMILES is that it is a unique notation. This was very useful during the graph rewriting process since every newly generated metabolite is checked against the entire metabolite pool for graph isomorphism. An exhaustive procedure of graph isomorphism checking with so many graphs would lead to an explosion in computational cost.

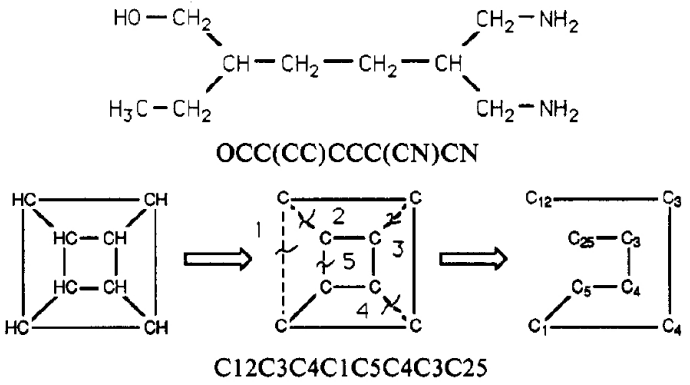


Figure 24: Examples for the generation of unique SMILES [Wei89]

For each graph we only have to generate its SMILES notation once and the graph isomorphism comes down to sequence comparison, which is drastically cheaper. In figure 24 two examples are shown, the upper one illustrating the branching problem and the lower depicting the generation of a unique smiles in case of a cyclic and completely symmetric molecule on the example of cubane.

To switch between the graph representation of a metabolite and its SMILES notation, a parser reading SMILES and generating the metabolite graph was implemented as well as a SMILES generator. To generate a unique SMILES notation, the graph nodes have to be assigned with canonical labels first. This is accomplished through the CANON algorithm[Wei89]. First, some graph invariants for each node are obtained; included are the number of connections of an atom, the number of non-hydrogen bonds, the atomic number and the number of hydrogen bonds. Based on these invariants, the atoms are sorted and ranked. If no two atoms share a rank, the labeling is done. Otherwise, the ranks are replaced by the product of primes corresponding to the neighbors' ranks and the atoms are sorted and ranked again. In some very symmetrical molecule graphs, it can happen that some atoms are always ranked the same, then this tie has to be broken and one atom has to be chosen to be in a lower rank. Uniqueness is still ensured. After the canonical labeling is derived, the final unique SMILES string is generated through a depth-first search through the metabolite graph[Wei89], starting with the node having the minimal canonic label. The depth-first search branches always to the node with the lower label, except in a ring it is branched toward a multiple bond if existing, this is basically done for the clarity of the notation.

4.2 Graph Rewriting

Graph rewriting is the process of applying rewrite rules, of the form $rule : G_{left} \rightarrow G_{right}$, with G_{left} being the left part of the rule and G_{right} the right part. Such a rule describes all possible transformations, $G_1 \xrightarrow{rule} G_2$, from a graph G_1 containing G_{left} to a graph G_2 for which G_{left} is replaced by G_{right}. In order to apply a rewrite rule, first G_{left} has to be matched to G_1. A check for subgraph isomorphism provides this matching if it exists. Following, the match in G_1 can be transformed to G_{right}, resulting in G_2.

$$\begin{array}{ccc} G_{left} & \xrightarrow{rule} & G_{right} \\ \Big| {\scriptstyle match} & & \Big| {\scriptstyle match} \\ \downarrow & & \downarrow \\ G_1 & \xrightarrow{rule} & G_2 \end{array}$$

4.2.1 Subgraph Isomorphism

A graph $G_{left} = (V_{left}, E_{left})$, with V_{left} being the set of vertices and E_{left} the edge set of G_{left}, is isomorphic to a subgraph S_1 of $G_1 = (V_1, E_1)$, with $S_1 \subseteq G_1$, if there is a mapping $m_v : V_{left} \rightarrow V_1$ and $m_e : E_{left} \rightarrow E_1$, so that holds $\forall v_i, v_j \in V_{left} : \{v_i, v_j\} \in E_{left} \rightarrow v_i =_{label} m_v(v_i)\ and\ \{m_v(v_1), m_v(v_j)\} \in E_1\ and\ m_e(\{v_i, v_j\}) = \{m_v(v_1), m_v(v_j)\} =_{label} \{v_i, v_j\}$.

The algorithm that is implemented in this simulation tool for subgraph isomorphism checking and finding all possible mappings of G_{left} in G_1 is based on a tree search with backtracking[KH04], which is a common procedure[CFSV04]. The current vertex and edge mappings are stored as secondary labels of vertices and edges of G_1, respectively. The tree search is performed as depth-first search, having the advantage that always only one mapping has to be in the memory. Thus, after finding one complete mapping, the corresponding rewrite rule is applied and the mapping discarded. The algorithm starts with an empty mapping and iteratively extends it with vertices and edges. It, first, tries to add a starting vertex and one further vertex and then looks for an edge connecting both vertices. Since a molecule graph is a labeled graph, the vertices and edges of the mapping have to have the same labels as those of the mapped graph. In every iteration, except of the last, another vertex and edge is added to the mapping, if they can be found in G_1. Otherwise, the algorithm backtracks in the tree, frees the vertex and edge which were added in the previous iteration. If the algorithm cannot find any mapping for the starting vertex anymore, then it stops and returns either true or false, depending on whether G_{left} is isomorphic to a subgraph of G_1 or not.

4.2.2 Rule Application

After finding a mapping of the left-hand side of the rule in the metabolite graph, the rule can be applied. This means, the part in the metabolite graph to which was mapped can be replaced by the right-hand side of the rule. In the case of chemical reactions, the transformation is solely concerning the edges of the respective area in the metabolite graph since during a chemical reaction only bonds change but the atoms stay the same. Since we do have an exact mapping of the edges in G_1 which correspond to G_{left} and the `graph` interface provides us with a function to set labels, we can simply relabel the edges, so that they contain the labeling of the bonds of G_{right}. If edges of G_{right} were not present in G_{left} or G_1, respectively, then those have to be added with the corresponding methods of the `metabolite` class. On the other hand, edges of G_{left} not contained in G_{right} are labeled 0. After this procedure is done, all edges with label 0 are discarded from the metabolite graph G_2. This is necessary, because for bimolecular reactions pairs of metabolite graphs have to be joined, which is realized by adding edges labeled with 0 between all pairs $\{v_{m1} \in G_{m1}, v_{m2} \in G_{m2}\}$. The reaction graph, G_{left} of a bimolecular reaction has two non-adjacent gaps, these are also labeled with 0. Consequently, the subgraph isomorphism check can be performed in the regular fashion, as described above. Since some of the edges in G_2 might have been removed, it is necessary to check whether the graph is still connected. While building the chemistry, it was ensured that there will be no chemical reactions involving more than two molecules, thus we can be sure that if a metabolite graph is not connected, that G_2 represents exactly two metabolites. The two connected components of the graph are obtained through a depth-first "flooding" algorithm. For each component, a new metabolite graph is generated and the corresponding vertices and edges are added. The actual application of the rule is now done, but whether the generated or transformed metabolites are actually added to the metabolite pool and the network graph, depends on whether they satisfy the requirements of the energy calculation and optional restrictions based on graph indices.

4.3 Toy Universe

Within the Toy Universe[Ben06], molecules are represented in another graph form -the orbital graph- which will be explained in the next section. From the orbital graph of a molecule all the necessary properties needed for the energy calculation can be derived. Once the orbital graph is derived, the actual energy calculation using a simplified extended Hückel theory (EHT) is performed. The `ToyChem` package provides even more possibilities than the energy calculation of a simple molecule, further features are the computation of solvation energies[BF04] and reactions rates[BF05a]. Solvation energies could be used to simulate environments of multiple

phases. Reaction rates lead to more realistic models of enzymes. It is intended to integrate these features in later simulation models.

4.3.1 Orbital Graph

In the orbital graph of a molecule, nodes are the atom orbitals and edges indicate overlapping orbitals. From the four atom orbitals $2p_x$, $2p_y$, $2p_z$ and $2s$, three hybrid orbitals with different geometry can be formed. The hybrid orbitals sp (linear geometry), sp^2 (trigonal geometry) and sp^3 (tetrahedral geometry) combined with the respective atom type constitute the node labels of the orbital graph. The edge labels depend on the orientation of the two interacting orbitals relative to each other. In `ToyChem`, three types are regarded. Therefore, there are three different edge labels, direct σ-overlap, semi-direct σ-overlap and π-overlap. In figure 25 an example orbital graph is shown to illustrate the composition of the nodes and edges.

4.3.2 Energy Calculation

The energy calculation in `ToyChem` is a simplification of the extended Hückel theory (EHT) [Ben06]. In quantum mechanics, electrons are described with the wave function Ψ satisfying the Schrädinger equation $\widehat{H}\Psi = i\hbar\frac{\partial\Psi}{\partial t}$, with $\hbar$ being the Planck constant and $\widehat{H}$ the Hamilton operator describing the total energy of the system ($\widehat{H}\Psi = E\Psi$). The molecular orbital Ψ_α is derived from the basis set of atomic orbitals $\{\chi_i\}$, $\Psi_\alpha = \sum c_{\alpha,i}\chi_i$, with $c_{\alpha,i}$ being the coefficients of the molecular orbital. The total energy of a molecule can be computed using the eigenvalues of the orbital energy E_α and the number of electrons n_α in the molecular orbital Ψ_α. $\widehat{H}\vec{c_\alpha} = E_\alpha S\vec{c_\alpha}$, with $S_{ij} = \int \chi_i\chi_j d\tau$ being the overlap matrix, and the total energy $E = \sum_\alpha n_\alpha E_\alpha$.

4.4 Topological Indices

A topological index or graph index, as will be referred to it sometimes throughout the thesis, is a number characterizing the constitution of a graph[Tri92]. The value of the index does not depend on the labeling of the graph or the way it is presented, thus, it can also be seen as a graph invariant. There are basically two groups of indices: one group consisting of those indices being based on the vertex or edge connectivity, the other containing indices which rely on distance information. Belonging to the first group, the Zagreb index and the Connectivity index will be introduced in the next sections and also are integrated as optional selections for chemical molecules in the simulation tool. From the second group the Wiener number, the Platt number and the Balaban index were selected. All indices are supposed to resemble a different chemical or graph-theoretical property.

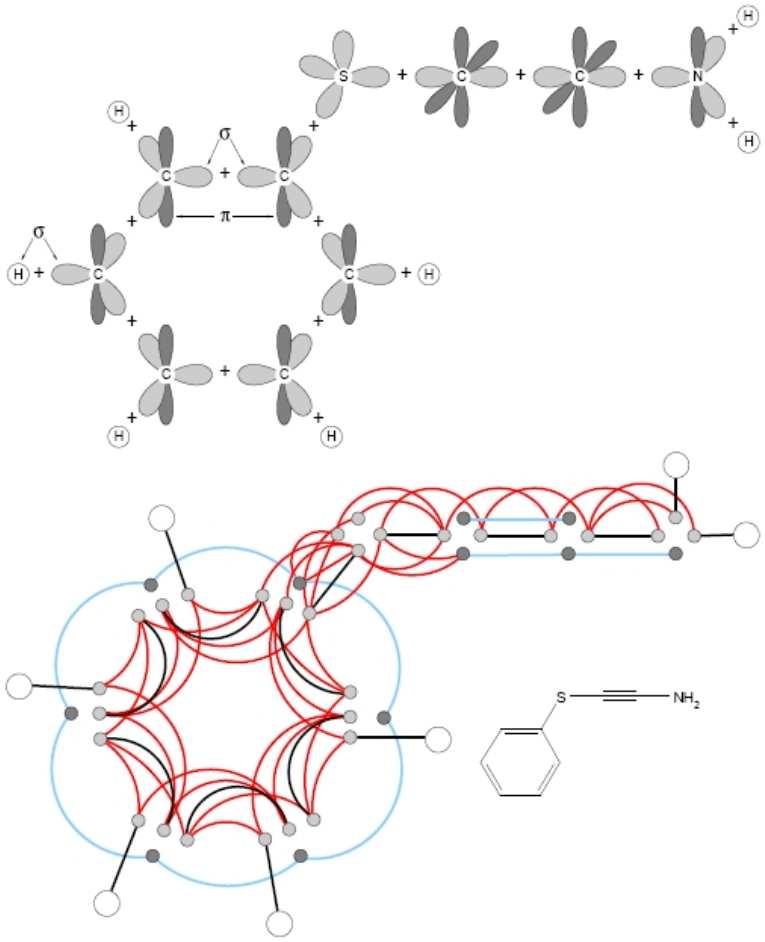


Figure 25: Orbital graph [Ben06]. Top: Decomposition of the molecule in hybrid orbitals. Bottom: Orbital graph as used for energy calculation. Nodes = orbitals, Edges: dark gray = p, light gray = sp^n and white = s arcs = overlaps between orbitals - black = σ-bonds, red = semi-direct, light blue = π-bonds.

Topological indices are interesting for Quantitative structure-activity relationship (QSAR), Quantitative structure-property relationship (QSPR) and other applications in the pharmaceutical industry[KH76]. QSAR, QSPR are methods to correlate the structure of a chemical molecule with its biological activity or property, respectively. Therefore, graph indices are the appropriate tools in these areas since they are simple descriptors that do not need empirically derived measurements and that are rapidly computed. This proves to be important for drug design which perform QSAR studies of billions of structures before the actual synthesis of the designed drug targets. With millions of such substances being developed every year, the simplicity of this method means a drastic improvement in research and development costs for pharmaceutical companies.

Within the simulation, the topological indices will be used as additional criteria for the selection of preferable reactions and metabolites besides the energy evaluation by the `ToyChem` package. So far, it is designed such that one of the indices can be optionally added as such a selection criteria, but there is no obstacle denying the possibility of combining indices. The introduction of topological indices to the model ought not to be understood as a biologically meaningful assumption of the importance of the corresponding chemical or graph-theoretical properties resembled by them. Rather, they will be used to study the behavior of the simulation itself and only to a small extent to make predictions about the nature of the property itself, e.g. whether it is an underlying property of substances in metabolic networks, or whether it can be seen only as a property distinguishing metabolites, or not have any biological meaning at all. Mainly, though, it will be interesting to see what kind of reactions will evolve for the different indices under the same starting conditions and whether it is possible to find correlations between the constitution of the reactions and the respective index. However, in future research more sophisticated indices might be integrated which then are supposed to lead to a biologically meaningful change in the type of metabolites evolving in the system. For example, it might be of interest to add an index for toxicity enabling to study how a system deals with such metabolites. Furthermore, this index could be combined with a set of other indices for properties of drug targets and thus possibly finding candidate structures inhabiting those desired properties but are, in fact, not toxic.

4.4.1 Zagreb Index

The Zagreb index is a topological index based on the connectivity[Tri92]. It was defined such that it correlates with the π-electron energy. There are actually two Zagreb indices: the original one, M_1, can be seen in equation 4, and the one used in the metabolite evaluation of the simula-

tion, M_2, in equation 5, with $D(i)$ being the valency of the vertex i.

$$M_1 = \sum_i D^2(i) \tag{4}$$

$$M_2 = \sum_{\{i,j\}} D(i) D(j) \tag{5}$$

4.4.2 Connectivity Index

The Connectivity index[Ran75] is calculated in a similar fashion as the Zagreb index. It also uses the valency of all vertices (equation 7). Despite the similarity between the two indices, the Connectivity index was supposed to be an indicator of molecular branching of a chemical structure, as opposed to the π-electron energy as is the case for the Zagreb index. However, it is also correlated with the π-electron energy of its metabolite. There are two more Connectivity indices that are commonly used. The original Connectivity index can be seen as of first-order and sums over all bonds in the molecule graph, the other two are the Connectivity index of zero order over all vertices(equation 6) and second-order over all paths of length two (equation 8). All three indices are shown at an example graph in figure 26.

$${}^0\chi = \sum_i [D(i)]^{-1/2} \tag{6}$$

$${}^1\chi = \sum_{\{i,j\}} [D(i) D(j)]^{-1/2} \tag{7}$$

$${}^2\chi = \sum_{\{i,j,k\}} [D(i) D(j) D(k)]^{-1/2} \tag{8}$$

4.4.3 Wiener Number

The Wiener number was one of the first indices based on distances and was first introduced as path number[Wie47], which is the number of bonds in a molecule. The Wiener number is calculated using the distance matrix D, as in equation 9. It is an indicator for the compactness of a molecule graph, as is illustrated in figure 27. In combination with other graph invariants, such as the polarity number p (equation 10 with p_3 being the number of paths with length 3), it can also indicate other physical properties of molecules, e.g. boiling point.

$$W = (1/2) \sum_k \sum_l (D)_{kl} \tag{9}$$

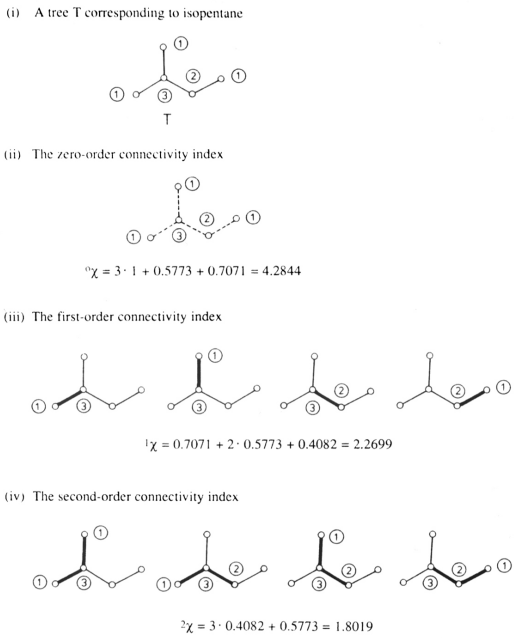


Figure 26: Calculation of ${}^0\chi$,${}^1\chi$ and ${}^2\chi$ on an example graph T [Tri92]

$$p = (1/2) \sum_i (p_3)_i \tag{10}$$

4.4.4 Platt Number

The Platt Number[Pla47], although, being a distance based index, is rather similar to the two indices described first. It is defined by the sum of edge-degrees in the molecule graph, as in equation 11 with $D(e)$ being the edge-degree of edge e. Since $D(e)$ is the number of adjacent edges of e, it can be easily obtained because the graph interface used in the simulation provides a function. Thus, it is not necessary to calculate it as in equation 12, where $D(i)$ is the valency of vertex i, which shows the similarity to Zagreb and Connectivity index. The Platt number was intended as a measure for molar volume and some other physical properties as the heat of formation or vaporization.

$$F = \sum_i D(e_i) \tag{11}$$

$$F = \sum_{\{i,j\}} [D(i) + D(j) - 2] \tag{12}$$

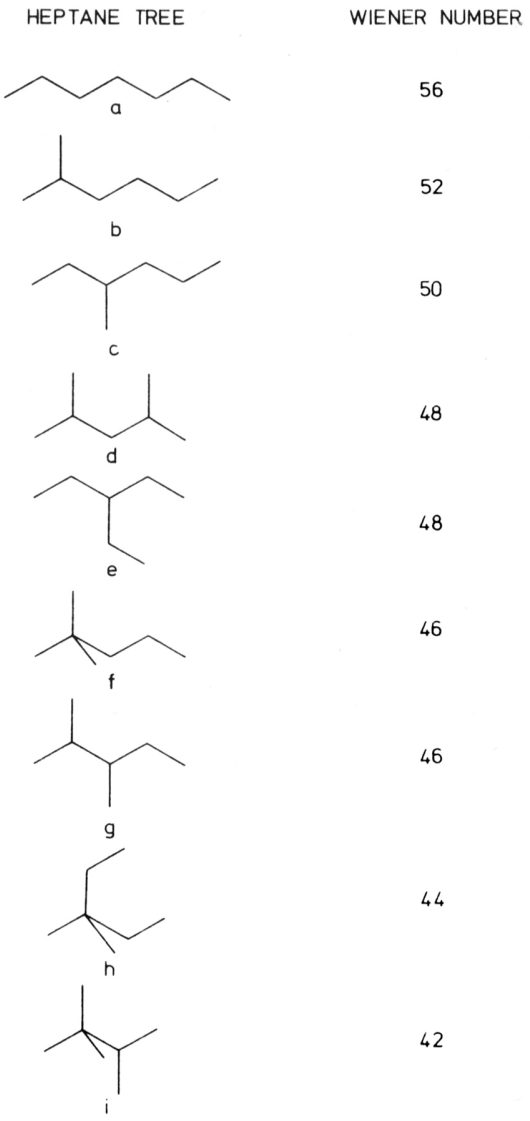


Figure 27: Ordering of heptane trees based on their Wiener number [Tri92]

4.4.5 Balaban Index

The Balaban index is defined as the average distance sum connectivity[Bal82]. It is calculated by equation 13, where d_i is the distance sum for vector i to all other vertices, M is the number of edges and μ is the cyclomatic number, thus, $\mu = M - N + 1$ with N being the number of vertices in the graph. For the sample graph in figure 28, the Balaban index can be computed from $M = 9$, $N = 8$, $\mu = 2$ and the distance sums $d_1 = 14$, $d_2 = 16$, $d_3 = 16$, $d_4 = 14$, $d_5 = 12$, $d_6 = 16$, $d_7 = 16$, $d_8 = 12$, resulting in $J = 1.9215$. The distance sums itself can be used as index being a measure for compactness of the respective area of the vertex. The Balaban index, however, is an indicator for the ramification of the molecule graph.

$$J = \frac{M}{\mu + 1} \sum_{\{i,j\}} (d_i d_j)^{-1/2} \tag{13}$$

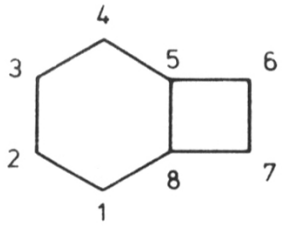


Figure 28: Example graph for the calculation of the Balaban index [Tri92]

5 Metabolic Flux Analysis

5.1 Introduction

Metabolic flux analysis is the calculation and analysis of the flux distribution of a steady-state metabolic network.

5.1.1 General Motivation

Since the genomic data available increased so tremendously in the last decade it became possible to reconstruct entire metabolic networks of many organisms. Knowledge about metabolic networks and fluxes proves to be important for functional genomics and bioengineering[Wie02, Sch99]. The interpretation of gene chip data often is combined with insights about the metabolic fluxes in the system of investigation, thus new patterns and relations among the genes may be discovered. For many years it is desired to increase the metabolic yield of organisms which are used in pharmacy or other industries. For such a goal, knowledge about structural properties is needed rather than the exact kinetic details of the system. Therefore, metabolic flux analysis seems to be the appropriate tool for the efforts made in the area of biotechnology, e.g. recombinant DNA technology.

The primary objective of metabolic flux analysis is apparently to gain insights about the metabolism itself. Metabolic flux analysis allows to make predictions about the flexibility, redundancy and robustness of metabolic networks[SHWF02, PSVN$^+$99]. Furthermore it can be used to identify novel pathways, or extremal pathways, e.g. pathways of optimal yield or infeasible cycles in a network[GK04]. It can also be determined whether reactions are correlated, i.e. always occur together in the same fluxes (enzyme subsets), and how important they are for the entire metabolism regarding the robustness or the yield.

5.1.2 Motivation in the project

Metabolic networks are also an important part of the simulation introduced in this thesis, thus analyzing them is of utter interest. The simulation starts with a population of individuals each containing a metabolic network. In each generation a selection of the fittest individuals is performed. Fitness in this context is usually considered a property of the whole network. Metabolic flux analysis enables us to calculate the fitness of individuals, providing many choices ranging from robustness to yield. In the beginning, it will be focused on the yield as selection criteria, but for future research, other properties will be of interest and thus, will be investigated.

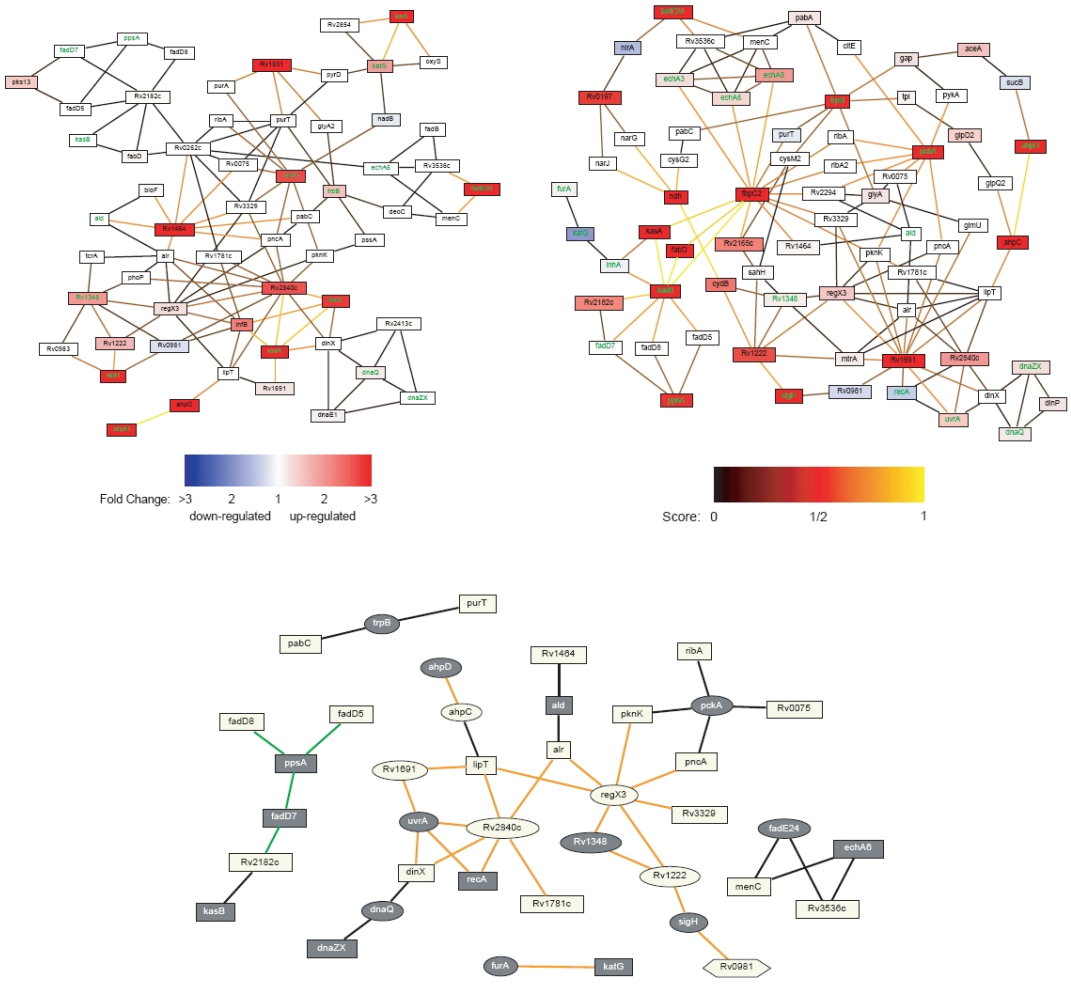


Figure 29: Building of response networks. Nodes and Edges based on metabolic information, coloring due to expression levels in gene chip. Top: 2 Response networks, Bottom: Difference of both networks. [CSFF05]

5.1.3 Stoichiometric Matrix

A stoichiometric matrix can be regarded as representation of a metabolic network with its reactions, metabolites and coefficients[UW05a]. Hence, such a matrix will be of size $\#reactions \times \#metabolites$ and have integer values as entries. In particular, the stoichiometric is defined as following, each column of the matrix represents one reaction of the metabolic network and each row a metabolite. Equation 14 is an example of a stoichiometric matrix and below is the interpretation of a column and a row vector respectively. It is easy to see how the network information is inhabited in the matrix and can potentially be regained. From a row vector we can determine in which reactions a certain metabolite is produced or consumed respectively, and to what extent. A column vector on the other hand tells us exactly which metabolites are involved in the reaction and in which role, thus giving us more than just information about the network structure but also holding chemical information by obeying chemical balance.

$$S = \begin{array}{c} \begin{array}{ccccccc} R1 & R2 & R3 & R4 & R5 & R6 & R7 \end{array} \\ \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \end{array} \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \tag{14}$$

Row 1: Metabolite A is produced by reaction R1 and consumed by R2 and R3
Column 2: Reaction R2 uses metabolite A t build B

The stoichiometric matrix can also be looked at from a mathematical view[Pal06], as linear transformation from the space of fluxes into the space of time derivatives of the metabolite concentration. Thus, given a certain flux vector v with v_i being the netrate if the i^{th} reaction we can determine how the concentration of the metabolites will change over time. Equation 15 clarifies the procedure.

$$\frac{dx}{dt} = s_1 v_1 + s_2 v_2 + ... + s_n v_n \tag{15}$$

5.1.4 Subspaces of S

For the stoichiometric matrix as for all matrices, there are four spaces which can be build from it[Pal06]. One space is formed by the columns of S and one by the rows of S, namely Col(S) and Row(S) respectively. The former containing the time derivatives, the flux vectors being contained in the latter. To both spaces exist corresponding null spaces, Left null(S) being connected

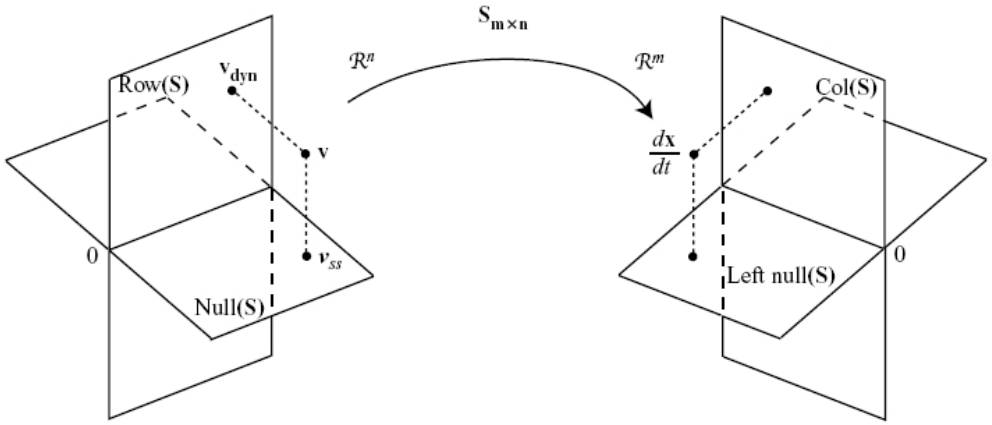


Figure 30: Subspaces of the stoichiometric matrix [Pal06]

to Col(S) and Null(S) to Row(S).

As one can tell from the name metabolic flux analysis, the focus will be on the space of fluxes. The vectors in this space contains two parts, a dynamical and a so called steady-state. Biologically, one may assume the metabolites in a metabolic network to be in equilibrium, due to the fact that metabolism has a fast processing of its components compared to other biological networks. This means that none of the metabolites accumulates or dissolves in the network, or, in other words, the concentration of a metabolite does not change for any given flux vector. Thus, following equation has to be fulfilled ($Sv = 0$). The network is then considered to be in steady-state. Only in stationary condition of the network is it possible to find pathways or cycles contained in it. Consequently, in metabolic flux analysis we also want to assume the condition of steady-state and therefore only consider the space Null(S) as possible result space.

5.1.5 Flux Modes

Biologically speaking flux modes are representations of pathways in the metabolic network. Mathematically they are vectors of length q, the entries being the netrate of the respective reaction (v_i - netrate of i^{th} reaction). These vectors are in relative proportion, which means that two vectors are equivalent if there is a scalar α so that $v = \alpha \times v'$. Additionally flux modes are restricted by the assumption of steady state, $Sv = 0$, and the irreversibility inequalities, $v_i \geq 0$ if i is irreversible.

Given these properties, all possible flux modes form a convex polyhedral cone in the flux space. This cone can be represented by the following set of flux modes P, with P as in equation 16

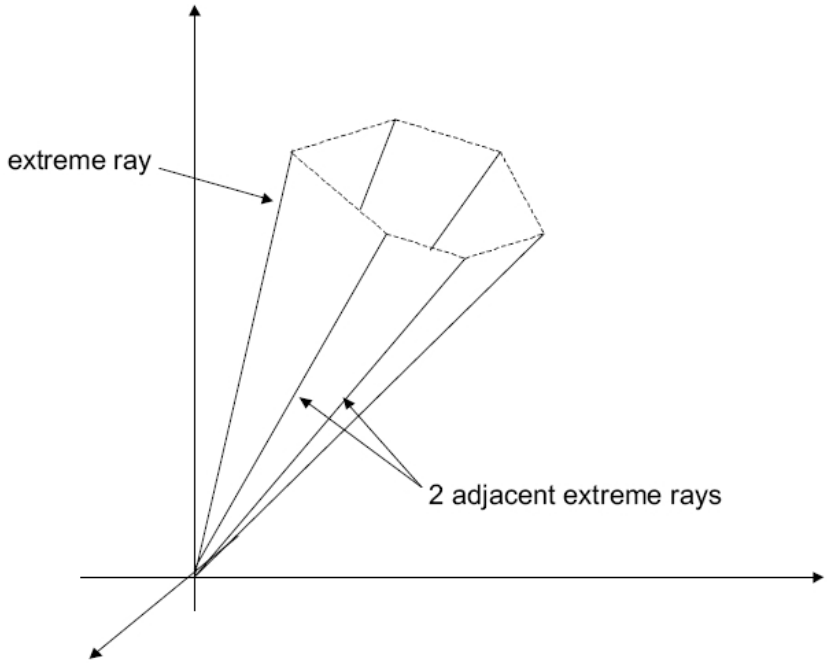


Figure 31: Pointed polyhedral cone [GK04]

$$P = \{v \in \Re^q : Sv = 0 \ \ and \ \ v_i \geq 0, i \in Irrev\} \tag{16}$$

Elementary Modes An elementary mode is a flux mode, hence a vector of length q, with the following property of elementarity[SSF]. A vector v is an elementary mode if it is contained in P, thus obeying the steady state criteria and the irreversibility constraints and there is no other vector v' in P being involved in fewer reactions than v. To be more precise, if both vectors v and v' are in P and v is elementary, then the set of reactions in which v' is involved must not be a proper subset of the reactions of v.

$$v \in P, \forall\, v' \in P: \ R(v') \ \subseteq \ R(v) \ \Rightarrow \ v' = 0 \ or \ v \simeq v' \ or \ v \simeq -v' \tag{17}$$

$$R(v) = \{i : v_i \neq 0\} \tag{18}$$

The set of elementary modes is a finite set of flux vectors which describes the infinite set P of possible fluxes in the metabolic network. Hence, each flux vector in P can be generated through linear combination of certain elementary modes and also each combination of elementary modes lies within P.

Based on the character of the set of elementary modes, we can deduce their use in the analysis of the metabolic network[UW05a]. First of all they are representations of pathways likely to be of importance because they are minimal routes through the network. But more expressiveness lies in the set of elementary modes. For example, it can be assumed that the number of elementary modes gives rise to network measures such as flexibility or stability. If there are many minimal routes, it is very likely that the network is stable against knockouts of reactions or at least is able to find alternative routes, hence, is flexible. Particularly obvious becomes the use of elementary modes if determining the optimal yield of the network or the pathways with optimal yield in the network, i.e. a maximal ratio from product to educt. Since all possible fluxes are linear combinations of elementary modes, it is clear that no non-elementary flux can be more optimal than an optimal elementary flux. Further it can be said that all combinations of optimal elementary modes are optimal fluxes as well. From these observations we can follow that the optimal yield of the network corresponds to the yield of the optimal elementary mode or modes. Another apparent use is to make statements about reactions. It can either be said something about single reactions, e.g. that a reaction is essential if it is involved in many elementary pathways, or statements can be made about the relation between reactions. If reactions always occur in the same elementary flux vectors than they can be considered a correlated subset or ,in terms of metabolic analysis, enzyme subsets. There are also other correlations conceivable, like sets of reactions which never occur together or work against each other.

Extreme Rays Extreme rays or extreme currents have very similar properties as elementary modes and in networks with only irreversible reactions they are even equal to them. The idea of extreme currents actually occurred earlier in the history of metabolic flux analysis but is discussed here following elementary modes. The reason is that elementary modes were used in the original null-space approach by Wagner[Wag04] and extreme rays later by Gagneur[GK04] while introducing their binary NSA, connected with the mathematical framework for enumerating extreme rays.

Not surprisingly the definition of extreme rays looks similar to the one in the section above. As the measure for the extremity we will use a different concept though. Instead of the set of involved reactions a so called Zero set Z is defined here. Z is a set of indices, where the index

indicates that the flux satisfies the respective inequality constraint with zero. The meaning of this definition and the distinction to the definition of elementary modes will become clearer in one of the next sections (section 5.1.7)in which the null-space approach will be discussed in detail. For now, the given explanation has to satisfy to define extreme rays. A vector v which is contained in P is an extreme ray if there is no non-null vector v' in P, so that the zero set of v is a proper subset of the zero set of v'. In other words v' must not satisfy the same inequality constraints as v plus some additional.

$$v \in P, \forall\, v' \in P : \; if\; Z(v) \subseteq Z(v') \;\; then\; v \simeq v' \tag{19}$$

$$Z(v) = \{i : \; A_i v = 0\} \tag{20}$$

The set of extreme rays is again a generating set for P and consequently, for all possible fluxes through the network. As opposed to elementary modes, a vector in P can be obtained only by non-negative linear combinations of extreme rays. Besides, we can only verify for non-negative combinations to be vectors of P. Therefore, P is not only a convex polyhedral cone, but also a pointed one.

Mathematical approaches for the pointed convex polyhedral cone exist, one of which is discussed in the next section.

5.1.6 Double Description Method

The Double Description Method[Mot53] is the most common method for enumerating elementary modes or as in this case extreme rays. To understand the Double Description method it is helpful to picture extreme rays as the edges of the pointed convex polyhedral cone satisfying all of the m equality constraints and $|Irrev|$ inequality constraints. It can simply be concluded from the convex properties of the extreme rays, that the rays of the cone satisfying m equality constraints are either rays of a cone satisfying $m-1$ of the constraints or they are linear combinations of rays of the $m-1$ cone. Consequently, there is a way to compute the extreme rays and, accordingly, the cone fulfilling m constraints from the extreme rays fulfilling only $m-1$ constraints. Starting point of our iterative method now could be the extreme rays of a cone satisfying $m = 0$ constraints, which would correspond to the entire $\Re^n$ and its edges being the vectors of the standard basis.

The idea behind the Double Description method is that we have two representations of the flux cone P, one describing P through the equality and inequality constraints (matrix A, equation

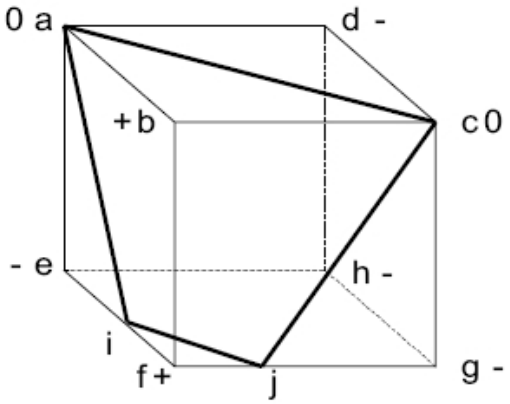


Figure 32: Double Description method [GK04]

21), the other defining P through its edges, the extreme rays (matrix R, equation 22). These two representations build the Double Description pair.

$$P(A) = \left\{ v \in \Re^d : \; Av \geq 0 \right\} \tag{21}$$

$$P(A) = \left\{ v \in \Re^d : \; x = R\lambda \; and \; \lambda \geq 0 \right\} \tag{22}$$

Corresponding to the considerations above, the method first needs a starting pair A and R both representing a cone satisfying a number of constraints, e.g. $m = 0$. At each step the matrix R represents the edges of the cone formed by the constraints in A. Iteratively, remaining constraints are introduced which reshape the cone and thus changing the set of extreme rays. Only if an extreme ray fulfills a newly introduced constraint it is kept. New rays are generated through linear combinations of existing ones, those of the preceding step. In order to form a valid new extreme ray which satisfies the new constraint, two extreme rays have to be adjacent, meaning that the resulting vector is not a linear combination of extreme rays of the new cone. Using the notion of Zero set, introduced in the last section, adjacency is defined in the following way: two rays v and v' are adjacent if there is no ray v'', so that the intersection of the Zero sets $Z(v)$ and $Z(v')$ from v and v' is a proper subset of the Zero set $Z(v'')$ from v''. After introducing all constraints in A, matrix R, in particular, the column vectors of R will represent the extreme rays of the final flux cone.

$$v, v' \in P, \; \forall \, v'' \in P : \; if \; Z(v) \cap Z(v') \subseteq Z(v'') \; then \; v \simeq v'' \; or \; v' \simeq v'' \tag{23}$$

5.1.7 Null-Space Approach

In this section we will discuss how the double description method is realized in the Null-Space approach[UW05b, UW05a, GK04]. As mentioned above the simple idea for choosing a starting double description pair would be a matrix A representing the full space of fluxes without any equality constraints ($Sv = 0$) and correspondingly a matrix R containing the respective extreme rays which in this simple case are the vectors of the standard basis of the $\Re^n$. So in the simple approach we first ignore the equality constraints and start with the full space of fluxes and then iteratively introduce one equality constraint after another. This would correspond to intersecting the full space with hyperplanes. The intersection of all hyperplanes forms the polyhedral cone.

The Null-Space Approach starts from another angle. First, it ignores all inequality constraints ($v_i > 0$ if i is irrev) but satisfies all equality constraints by calculating a certain Kernel Matrix K of S ($S \times K = 0$) and using this K in the starting double description pair. To visualize this idea again, by starting with the kernel matrix, we restrict our search to the space Null(S) in the first place. Remember from the section subspaces of S (5.1.4) that we want to observe only Null(S) because we assume the network to be in steady state. Now instead of intersecting the space with hyperplanes representing the equality constraints, we intersect the null space of S with half-spaces each corresponding to an inequality constraint.

Both approaches obviously yield to the same result, the same cone is formed and the same extreme rays obtained. The advantage of the null-space approach now lies in the computational part. The cost of calculating the kernel matrix in the beginning is negligible compared to the remaining extreme ray enumeration but it has the benefit that we can limit our search space from the full-space to the null-space. Another advantage becomes apparent through the following observation: when intersecting the search space with half-spaces instead of hyperplanes it is likely that we can keep more extreme rays from the preceding iteration, because a ray is more likely to fulfill an inequality constraint ($v_i \geq 0$) than a more restrictive equality constraint ($Sv = 0$). Since we obtain the same number of extreme rays at the end and discard fewer rays during the iteration steps, it follows that we also have to compute fewer new rays and perform fewer adjacency tests which usually takes up most of the computation.

The kernel matrix K of S, serves as the part of the matrix R in the initial double description pair. The column vectors of K are linear independent and represent the extreme rays of the null-space. Before applying the double description method, K is brought into a specific structure, which will be explained in the Methodology section (5.2). The matrix A will be of such a

form that it describes the null-space without any inequality constraints. It is again referred to the Methodology section (5.2) for further details.

5.1.8 Binary Approach

The intention of the binary approach is to store part of the matrix R in a binary presentation instead of real numbers in order to be more memory efficient[GK04]. Since R can grow dramatically in the course of the enumeration and takes up most of the memory, it is obvious that a binary presentation would be preferable. As a pleasant side-effect; it also has an advantage computation-wise because adjacency tests in the binary approach come down to simple bit-operations. Those are faster than real number vector operations and also numerically exact.

The reason for the possibility of using the binary presentation rather than real numbers is twofold. Firstly, for the adjacency tests of two vectors in the null-space approach only the entries of already processed rows are needed. In the first iteration this means the first $r - m$ rows of the kernel matrix. As discussed above, K contains of a identity matrix of size $r - m$ in the top. Hence all entries needed for the adjacency tests in the first iteration are either 0 or 1, so basically already of binary fashion. Secondly, if two rays are found to be adjacent and a new extreme ray is generated, than it will be a positive linear combination. Therefore, we cannot obtain a negative value, but only values equal to zero or above zero. Looking at the definition of adjacency from previous sections, we notice that this is the only information needed for the decision.

The part of R which is in binary presentation consists of all the already processed rows. Thus, the fraction of the binary part compared to the real number part becomes bigger with each iteration. The final matrix R and, consequently, the extreme rays are completely binary. While for many applications the binary form of the extreme rays suffice, some other applications, in particular, the determination of the optimal yield, demand the real number presentation. Fortunately, there is an easy way to reconstruct the real number vector of the binary extreme ray. Only the following equation has to be solved.

$$S_{R(v)} \times v_{R(v)} = 0 \tag{24}$$

Where $S_{R(v)}$ is the submatrix of S containing only reactions involved in extreme ray v and $v_{R(v)}$ is the vector containing all non-zero entries of v. The solution-space of this equation is always

one-dimensional if v is an extreme ray. Now only one coefficient of the solution has to be fixed and all other can be derived. The vector can be normalized if desired.

5.1.9 Network Reduction

Although ideas like the null-space and the binary approach have helped to decrease the time and memory complexity of the extreme ray enumeration algorithm, it would be still desirable to make further improvements, in order to be able to perform metabolic flux analysis on even larger networks in reasonable time and with manageable resources. In this section some ideas, which can be summarized as forms of network reductions, are being discussed. These are not improvements of the algorithm itself but rather preprocessing steps to decrease the problem size the algorithm has to solve. Since the outcome of the algorithm must not be different, these approaches cannot avoid the exponentially increasing number of extreme rays, but they may reduce the number and the complexity of adjacency tests being performed.

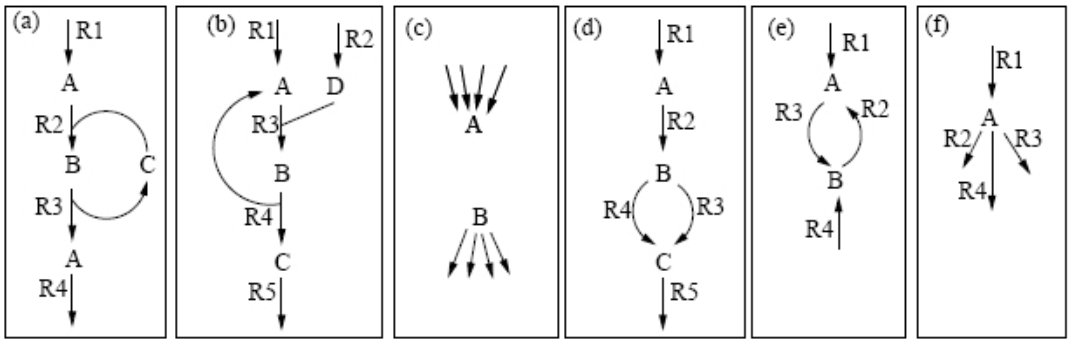


Figure 33: Examples of network redundancies [GK04]

In this section we will discuss the different reductions on the network level, in the Methodology section (5.2) we will go further into the algorithmically realization.

Conservation Relations The first approach is the reduction of conservation relations. A conservation relation is present if rows of the stoichiometric matrix, i.e. vectors of metabolites, are linearly dependent. An easy example can be seen in figure 33a where only two metabolites are involved in the conservation relation, namely metabolite B and C. The row vector of metabolite C does not add any new information for the flux analysis and, therefore, will be discarded.

Strictly Detailed Balanced Reactions The next reduction is the one of strictly detailed balanced reactions[SS91]. These are basically reactions with null flux in steady-state. The easiest examples of such reactions are those which flow into a sink of the network or out of a source. This can be seen in figure 33b. With sink and source of a network a metabolite that has only incoming or only outgoing reactions is described, respectively. Another example is case c in figure 33, where reaction R1 would be classified as strictly detailed balanced reaction. This reaction has a null-flux at steady-state because the metabolite A, in which it is flowing, is already inevitably maintained by the cycle $A \rightarrow R3 \rightarrow B \rightarrow R4 \rightarrow A$. Consequently, whenever A has an out-flux, it will have an influx of the same amount. Therefore, it is impossible for R1 to be involved in a steady-state flux because its participation would lead to a concentration of A greater 0. In this case the respective reaction R1 would be discarded. In the case of a sink and source the metabolites would be ignored for further analysis as well.

Enzyme Subsets Another type of reactions which has to be looked at is enzyme subsets. Reactions in an enzyme subset are always participating in the same fluxes and to similar extent, i.e. their ratios are always equal in the sense that $v_{1i} = \alpha \times v_{2i}$ for all reactions i. If α is greater or equal 0 than we can keep one of the reactions from the enzyme subset and discard the others. This reduction is possible because we can say that if the remaining reaction occurs in an extreme ray, the other discarded reactions must also participate in this flux and so we can easily reconstruct the extreme ray of the unreduced network. Figure 33d shows an example of that, reactions R1, R2, R5 being one enzyme subset. If α is below 0 on the other hand, we have to discard both reactions. The reason for that is following: since both reactions are active at the same time and both flux-ratios are opposite to each other, a flux vector containing these reactions cannot fulfill the inequality constraints of both of them. Therefore, it will not lie in the polyhedral flux cone and, consequently, can not be an extreme ray. If both reactions are not active, then obviously their flux is zero and they are not interesting for further analysis. An example for that can be seen in figure 33e, where R1 and R4 are opposite to each other.

Other Reductions There are some further ways of reducing the problem size of the enumeration. One is similar to the reductions discussed above. It looks for metabolites which have only one incoming reaction and several going out, or the opposite case having only one outgoing reaction and several coming in. Now this metabolite will be eliminated from the network and the single reaction will be combined with all other reactions on the opposite side of the metabolite. In figure 33f an example for such a case is illustrated, where A would be the metabolite we want to ignore and R1 the reaction we would have to combine with R2, R3 and R4. In this implementation, one other reduction step is performed, but it is of more technical nature and rather

difficult to explain in terms of the network topology. Thus, we will leave the explanation to the Methodology section (5.2).

A completely different approach of reducing the network size, is splitting the network into different subnetworks[SPM+02]. Based on some properties of reactions or metabolites one can determine with a certain probability the border between two subnetworks. For example reactions which have lowly connected metabolites as input or output are thought to be essential connectors between subnetworks[SS06]. A splitting of subnetworks would correspond to the divide and conquer approach known to be useful for algorithms in computer science. It makes especially sense since the complexity of the extreme ray enumeration grows exponentially with the size of the network. However, network splitting is not applied in this approach.

5.1.10 Yield Determination

When one analyzes metabolic networks and wants to compare them in an evolutionary sense it seems obvious to look at the metabolic yield of each network. A cell with a metabolism that produces more useful metabolites or at a higher rate will be evolutionary fitter and is more likely to be picked by natural selection. That is, if a cell reaches a certain level of metabolites responsible for the growth of the cell, it will split and if it reaches this level faster, it can multiply more often and, consequently, supersede slower growing cells. How the goodness of a metabolite can be assessed is discussed in the section Topological Indices (4.4). The fitness of a metabolic network is a combination of the optimal yield of the network and the goodness of the metabolites involved in the pathways of optimal yield.

Overall Stoichiometry Before we can compare metabolic networks through a fitness value or even determine their optimal yield, we have to derive the overall stoichiometry based on the extreme rays. For this process and also the following steps we need to define two sets of metabolites, external and internal metabolites. Further we distinguish two kinds of external metabolites. Some will serve as input for the internal network, which would usually be ubiquitous metabolites of the environment. Others will be defined as the output of the internal network. In principle, this could be metabolites having a strong effect on the growth of a cell. In this approach, some of the best scoring metabolites will be picked. Again see section 4.4 for further explanation. The internal metabolites are all the remaining metabolites.

Since the external metabolites are not included in the process of the extreme ray enumeration they do not underlie the restriction of steady-state. Therefore, their concentrations may change over time and one can calculate the changes for a given flux. This will be done here for all of

the extreme rays, in the methodology part it will be discussed how it is implemented into the algorithm.

Optimal Yield Given the stoichiometry for each extreme ray, it is now easy to determine the optimal yield of the network, i.e. the maximal ratio of product to educt of all fluxes within the metabolism. We know already that each flux through the network must be a combination of extreme rays, therefore, the yield of a combination can never be higher than any yield of the combined fluxes. This means that at least one of the extreme rays must be an optimal pathway and have optimal yield. Consequently, we only need to find the maximal ratio of product to educt among the set of extreme rays.

5.2 Methodology

In this section we will explain how the basic framework of the binary null-space approach[GK04] is realized in this implementation. For better understanding, we will accompany the explanations with an example. The sample network that we will use throughout this section will be presented as graph, see figure 34, where arrows represent reactions and indicate their (thermodynamically) direction and nodes stand for the metabolites of the network. Both nodes and arrows are labeled. The internal presentation of the network, the intermediate stages and the set of extreme rays are matrices and, accordingly, will be illustrated as such in the form of tableaus. Equation 14 is the matrix corresponding to the sample network from figure 34.

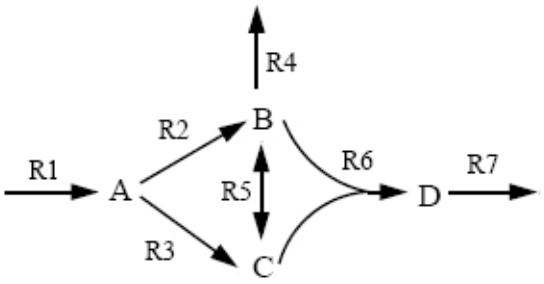


Figure 34: Example network graph [GK04]

5.2.1 Network Reduction

Before we come to the actual algorithm to perform the extreme ray enumeration, we will proceed with the preprocessing steps, namely the different network reductions, discussed earlier.

The preprocessing is performed in two major routines which for themselves consist of some other minor steps. Here we will not explain how to derive the stoichiometry matrix and kernel matrix. Explanations for the latter will be given later in the Methodology section. For the former knowledge from earlier chapters is assumed.

In the Introduction part of this chapter the several reduction types and their treatment were discussed separately. However, as we will see in the following sections, in the actual implementation sometimes reductions of the same type are handled in different steps and, on the other hand, reductions of different types are sometimes processed in the same step.

Reduction of the Stoichiometry Matrix The first preprocessing step analyzes the row vectors of the stoichiometry matrix of the original network, i.e. the network is still containing the reversible reactions. We distinguish three different cases. The first two observe all row vectors separately, whereas, the last case regards the stoichiometry matrix as a whole.
A metabolite is considered an internal source or sink if its row vector contains no positive or no negative entries, respectively. Since we still have the reversible reactions included the row vector may not have non-zero entries for the reversible reactions. If such a row is found, all involved relations, i.e. all columns with non-zero entries in this row, are discarded. The row itself can also be eliminated from further analysis. This step finds some strictly detailed balanced reactions, but some other more complex cases have to be found otherwise.

For the other case the preprocessing tries to find metabolites that are uniquely produced or consumed, i.e. they either have only one incoming or only one outgoing reaction. Again, each row vector is checked separately for a certain pattern of entries. Now we look for rows which have only one positive (negative) entry and several negative (positive) entries. The entries have to correspond to irreversible reactions. First the column with the single positive (negative) entry is combined pairwise with all columns having a negative (positive) entry in this row. With a combination a simple vector addition is meant.Consequently, the former vector is added on to each of the latter described vectors. Finally the single column vector as well as the row vector is discarded from the enumeration process. This step realizes one reduction discussed in section 5.1.9 completely and dissolves some enzyme subsets.

The last case is actually not a reduction step in that sense, it is a necessity which derives out of the calculation of the kernel matrix. When forming the kernel matrix, the stoichiometry matrix undergoes a gaussian elimination. If a matrix contains linear dependent rows gaussian elimination will result in a matrix containing a number of null rows, discarding these null rows makes

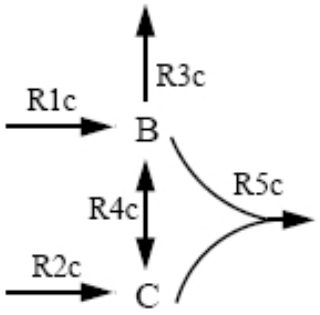


Figure 35: Reduced example network graph [GK04]

the rest of the matrix linear independent. The same is true for the kernel matrix computing, in the course of this step all linear dependent rows of S are eliminated. This corresponds to the elimination of the conservation relations discussed earlier. The final reduced stoichiometric matrix S_C is given in equation 25 and the corresponding network graph is depicted in figure 35.

$$S_C = \begin{array}{c} \begin{matrix} R1_C & R2_C & R3_C & R4_C & R5_C \end{matrix} \\ \begin{pmatrix} 1 & 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & 1 & -1 \end{pmatrix} \end{array} \begin{matrix} B \\ C \end{matrix} \tag{25}$$

Reduction of the Kernel Matrix The reduction of the kernel matrix comes down to one single step, the elimination of linear dependent rows of the kernel matrix. It should be mentioned that rows now correspond to reactions and columns to pathways. Accordingly, an entry (i,j) in the kernel matrix indicates whether a reaction i is involved in a pathway or extreme ray j. Eliminating all linear dependent rows has several consequences. First of all, we get rid of null rows that are cases of strictly detailed balanced reactions, i.e. reactions that are involved in none of the extreme rays and, thus, in no pathway feasible for the network. Another effect is that we detect and dissolve the remaining enzyme subsets not found in the first preprocessing step. Two reactions belong to the same enzyme subset if their row vectors in the kernel matrix are equal up to a scalar, i.e. if $r_{1i} = \alpha \times r_{2i} \; \forall \; i$. If α is positive one of the rows is kept and the other discarded. However, if α is negative, both rows (reactions) are ignored for enumeration. For the last consequence there is no name, but we can state that it eliminates linear dependent reactions in the kernel matrix, i.e. row vectors that are linear combinations of other row vectors. In terms of the network this would correspond to something similar to the lumping of enzyme subsets. Hence, the treatment on the algorithmic level is the same as mentioned above for the pairwise linear dependency.

Back Transformation Since the enumeration is done on the reduced network, obviously the extreme rays are also reduced, i.e. contain only entries for the reactions of the reduced network. As already stated, the number of extreme rays is the same in the original and the reduced network. Still one wants to reconstruct the extreme rays to the original size. Thus, the entries of the previously discarded reactions have to be regained from the existing ones. In order to accomplish that, it is necessary to keep track of all reductions of reactions, reductions of metabolites are insignificant since we only want to know which reactions are involved in a certain pathway. The reconstruction is easier when done on the binary form of the extreme rays rather than the real number vectors. First of all, in case a reaction was combined with one or more reactions, only the reactions have to be protocoled and not the exact proportion. And secondly, the reconstruction of reactions depending on more than one reaction comes down to a binary OR operation.

For strictly detailed balanced reactions, we already know that their entries are zero for all extreme rays. Conservation relations only eliminate metabolites and, therefore, we don't have to handle this kind of reduction for the protocol. For enzyme subsets, we have to distinguish two cases depending on the sign of the scalar. If the scalar α, $v_{1i} = \alpha \times v_{2i} \; \forall \; i$, is negative, then the reactions have a zero entry in each extreme ray. If α is positive, than the entries of the reduced reaction are a combination (binary OR) of the other reactions in the enzyme subset. For reactions which were eliminated due to flowing into (coming out of) a uniquely produced (consumed) metabolite the latter case applies as well. It can be noted that all reductions of reactions can be reconstructed sufficiently for our purposes.

5.2.2 Binary NSA

In the Introduction section (5.1) we already discussed the justification of the binary null-space approach and started to explain the basic procedure, starting from the choice of the double description pair to the iterations. In this section it will be explained how exactly the matrices and the operations on them look like. Below is the pseudocode of the entire metabolic flux analysis tool.

1: $PreProcessing(S)$ {sec. 5.2.1}
2: $RowEcholon(S)$ {per Gauss elimination}
3: $KernelMatrixComputing(S, K)$
4: $PreProcessing2(K)$ {sec. 5.2.1}
5: $InitialTableau(K, R_1, R_2)$ {R_1=binary part of R, R_2=real number part of R}
6: **while** !R_2 isEmpty **do**
7: **for all** rows i in R_2 **do**
8: **if** $R_2[i][] < R_2[1][]$ **then**
9: exchange $R_2[1][], R_2[i][]$ {Greedy-like Row-order: minimize $|Neg| \times |Pos|$}
10: **end if**
11: **end for**
12: **for all** entries j in $R_2[1][]$ **do**
13: **if** $R_2[i][j] > 0$ **then**
14: $Pos \Leftarrow j$ {Introduction of an Inequality constraint}
15: **else if** $R_2[i][j] < 0$ **then**
16: $Neg \Leftarrow j$
17: **else**
18: $Zero \Leftarrow j$
19: **end if**
20: **for all** p in Pos and n in Neg **do**
21: **if** $Adjacent(p, n, R_1, R_2[1])$ **then**
22: $ray = LinearCombination(p, n, R)$ {Binary and Double}
23: $R \leftarrow ray$ {New Column is added to R}
24: **end if**
25: **end for**
26: **for all** n in Neg **do**
27: delete $R[][n]$ {All Columns in Neg are discarded}
28: **end for**
29: **for all** p in Pos and z in $Zeros$ **do**
30: add 1 to $R_1[][p]$ and delete $R_2[1][p]$ {Update R_1 and R_2}
31: add 0 to $R_1[][z]$ and delete $R_2[1][z]$
32: **end for**
33: **end for**
34: **end while**
35: $Binary2Double(R)$
36: $PostProcessing(R)$

Initial Tableau The main structure that is used throughout the algorithm and on which the iterations work on, is a matrix or tableau divided in a binary and real number part. In the following, it will be described how the initial tableau is derived from the network.

From previous sections the notion of double description pair and the idea behind it should be known. For the null-space approach the matrix A was chosen such that it represents the null-space of the stoichiometry matrix with the inequality constraints yet to be satisfied. At this point all reversible reactions are already split to two irreversible ones. Therefore, we know that $P(A)$ is a pointed cone. Accordingly, the matrix A of the starting double description pair has the form as can be seen in equation 26. If A is chosen in this way, the matrix R has to be a kernel matrix of the stoichiometry matrix being in the form as in equation 27. Through row operations, we can bring a kernel matrix in reduced row-echelon form and, thus, in the desired structure.

$$A_{q+m} = \begin{bmatrix} I_{q-m} & 0_{(q-m)\times m} \\ & N \\ & -N \end{bmatrix} \tag{26}$$

$$R = \begin{pmatrix} I_{(q-m)\times(q-m)} \\ \widetilde{K} \end{pmatrix} = \left(\begin{array}{ccc} 1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 1 \\ \hline r_{q-m+1,1} & \dots & r_{q-m+1,q-m} \\ \dots & \dots & \dots \\ r_{q,1} & \dots & r_{q,q-m} \end{array}\right) \tag{27}$$

In principle, the algorithm could now proceed with exactly this double description pair, but for computational reasons another step is inserted. The justification for the following step is that a kernel matrix of the starting pair with as many zeros as possible is likely to reduce the number of adjacency tests having to be performed. This can be achieved by doing the following: first, the kernel matrix of the network still including the reversible reactions is computed, so that as many reversible reactions as possible are contained in the kernel matrix, i.e. not in the $r - m$ identity of K. Further, the reversible reactions are split in two irreversible reactions as following: the original row is kept ,representing one direction, and a new row is inserted for the opposite reaction. If the original row was not contained in the identity than the case is easy and a column representing the two-cycle between the forth and the back reaction is inserted. This column vector only contains two one-entries at the positions for the two connected irreversible reactions

and it has just the effect that the identity increases by one. In the other case, we also insert a new column vector, but for this one the entries have to be derived just like it is done for other reactions and thus we would not achieve an increase in the number of zeros. In equations 28, 29 and 30 this procedure for computing the kernel matrix from the reduced and splitted network, as in figure 36, is illustrated. The example shows only the case in which the reversible reaction is below the identity. In general, for most reactions, it is possible to realize that. Equation 31 represents the derived initial tableau -R^4 in this case- since we have already four processed rows.

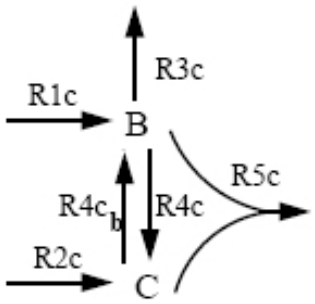


Figure 36: Reduced example network graph with splitted reactions [GK04]

$$S'_C = \begin{array}{c} \begin{matrix} R1_C & R2_C & R3_C & R4_{C_b} & R4_C & R5_C \end{matrix} \\ \begin{pmatrix} 1 & 0 & -1 & 1 & -1 & -1 \\ 0 & 1 & 0 & -1 & 1 & -1 \end{pmatrix} \end{array} \begin{matrix} B \\ C \end{matrix} \tag{28}$$

$$K_C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.5 & -0.5 & -0.5 \\ 0.5 & 0.5 & -0.5 \end{pmatrix} \begin{matrix} R1_C \\ R2_C \\ R3_C \\ R4_C \\ R5_C \end{matrix} \tag{29}$$

$$K'_C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0.5 & -0.5 & -0.5 & 1 \\ 0.5 & 0.5 & -0.5 & 0 \end{pmatrix} \begin{matrix} R1_C \\ R2_C \\ R3_C \\ R4_{C_b} \\ R4_C \\ R5_C \end{matrix} \tag{30}$$

$$R^4 = \left(\begin{array}{cccc} \times & 0 & 0 & 0 \\ 0 & \times & 0 & 0 \\ 0 & 0 & \times & 0 \\ 0 & 0 & 0 & \times \\ \hline 0.5 & -0.5 & -0.5 & 1 \\ 0.5 & 0.5 & -0.5 & 0 \end{array}\right) \begin{array}{l} R1_C \\ R2_C \\ R3_C \\ R4_{C_b} \\ \\ R4_C \\ R5_C \end{array} \tag{31}$$

The initial tableau now is build from the transformed kernel matrix. The identity of K is converted into a binary representation and then constitutes the upper part of the tableau. The remaining rows of K build the real number part.

Iteration The basic procedure for each iteration is simple and can be divided in two steps. The first one checks whether the existing extreme rays fulfill the new constraint and then either keeps or eliminates the respective column vector. The other step is the performance of the adjacency tests between pairs of extreme rays of the previous iteration to generate extreme rays of the new cone.

In each iteration one row of the tableau is processed. The binary part constitutes the already processed rows, which also means that in the initial tableau the first $r - m$ rows are already processed. Processing a row in the tableau corresponds to introducing a new inequality constraint. Since we have as many linear inequalities as reactions and $r - m$ of them are already satisfied in the initial tableau, we have a total of m inequalities to satisfy and thus m iterations to process. With m here is meant the rank of S and not necessarily the number of metabolites of the original network but rather of the reduced network. Due to reductions of the kernel matrix, m might be even smaller than this number.

The first step of introducing a new inequality constraint is to check for each extreme ray whether it satisfies the constraint. In terms of the tableau, this means that the entries of the current row that is to be processed in this iteration are all below or equal to zero. Columns in which this is the case are kept. Their entries in this row are brought in binary form and added to the binary part of the tableau. Accordingly, the real number part is decreased by this entry. Columns with negative entries in the current row are eliminated. Information about the signs of each column are kept for the adjacency tests.

An observation of experimental studies is that the order in which the rows of the tableau are processed has significant impact on the number of adjacency tests throughout the algorithm and,

thus, its computation time. So far only one approach for such an order of rows has been introduced and compared to the results when using the original order. The idea, suggested by Gagneur[GK04], is to order the real number part of the tableau according to the number of zeros in the rows, starting with the row with the most zeros. If a row has many zero-entries, it follows that fewer adjacency tests have to be performed since only extreme rays with positive entries are combined with those having negative entries in the respective row. This implementation follows this idea and even goes further. Instead of just considering the number of zeros, the exact number of adjacency tests that have to be processed is taken as criteria for the ordering. The product of positive entries and the number of negative entries in a row resembles exactly this value. Furthermoe, the idea is extended from simple sorting of rows to a greedy algorithm. That is, in every iteration the row with the minimal product is chosen.

After eliminating extreme rays not satisfying the newly introduced constraint, the next step is to generate the new extreme rays if existing. Recall from the Introduction section (5.1) that new extreme rays can only be generated from adjacent extreme rays of the previous iteration. We only have to consider pairs of rays where one ray satisfies the constraint and the other does not. It is obvious, that a combination of two rays which do not satisfy a constraint will not satisfy this constraint either. On the other hand, two rays fulfilling the constraint can by definition not be adjacent. We can restrict the set of pairs even further. The ray satisfying the constraint actually may not do so with equality. The reason for this is that the newly generated extreme ray lies on the hyperplane, representing the constraint, and the only linear combination of such a pair with this property is the extreme ray, satisfying the constraint with equality itself. Transforming this idea back to the tableau representation, all columns having a positive entry in the current row are checked for adjacency in combination with all columns having a negative entry. The adjacency test is realized as following: for each pair of extreme rays fulfilling the restrictions, discussed above, the binary-parts of both column vectors are combined to one binary vector through a binary OR operation. The combination of the entries of the current row also has to be included. Since they are combined in such a way that the new ray satisfies the inequality constraint with equality, this combination will always be zero. Consequently, the new binary vector will be extended with a zero-entry for this row. This vector is now checked against all extreme rays of the new cone, also including other newly generated extreme rays. If there is a vector among them containing all zero-positions of the combined vector, than it has failed the adjacency test and will not be added to the new set of extreme rays. Otherwise, the pair of extreme rays which were combined are considered adjacent and a new extreme ray will be added. The binary part of the new column in the tableau is already existent with the binary vector used for the adjacency test. The real number part is a positive linear combination of the real number vectors of the two

original columns. To which extent each column vector participates in the combination depends on the entries in the current row. Now the iteration is complete. Equations 32 and 33 show the two iterations performed for the example network of 34.

$$R^5 = \left(\begin{array}{cccccc} \times & 0 & \times & \times & 0 & 0 \\ 0 & 0 & \times & 0 & \times & 0 \\ 0 & 0 & 0 & \times & 0 & \times \\ 0 & \times & 0 & 0 & \times & \times \\ \times & \times & 0 & 0 & 0 & 0 \\ \hline 0.5 & 0 & 1 & 0 & 1 & -1 \end{array}\right) \begin{array}{c} R1_C \\ R2_C \\ R3_C \\ R4_{C_b} \\ R4_C \\ R5_C \end{array} \tag{32}$$

$$R^6 = \left(\begin{array}{cccccc} \times & 0 & \times & \times & 0 & 0 \\ 0 & 0 & \times & 0 & \times & \times \\ 0 & 0 & 0 & \times & 0 & \times \\ 0 & \times & 0 & 0 & \times & \times \\ \times & \times & 0 & 0 & 0 & 0 \\ \times & 0 & \times & 0 & \times & 0 \end{array}\right) \begin{array}{c} R1_C \\ R2_C \\ R3_C \\ R4_{C_b} \\ R4_C \\ R5_C \end{array} \tag{33}$$

Back Transformation After the last iteration the real number part of the tableau is empty and all rows are in binary presentation. It is now desired to transform each column back to its real number presentation. To achieve that, following equation 34 has to be solved. For that purpose, a submatrix of S and a subvector of v only containing the reactions having 1 as entry are generated. The submatrix is then brought into reduced row-echelon form. After setting one of the double values the other can be easily derived by gaussian backwards elimination. This is depicted for one flux mode of the example in equation 35.

$$S_{R(v)} v_{R(v)} = 0 \tag{34}$$

$$S_{C,R(v^1)} v^1_{R(v^1)} = \begin{array}{c} \begin{array}{ccc} R1_C & R4_C & R5_C \end{array} \\ \left(\begin{array}{ccc} 1 & -1 & -1 \\ 0 & 1 & -1 \end{array}\right) \end{array} \quad \left(\begin{array}{c} v^1_{R1_C} \\ v^1_{R4_C} \\ v^1_{R5_C} \end{array}\right) = \left(\begin{array}{c} 0 \\ 0 \end{array}\right) \tag{35}$$

5.2.3 Yield Determination

Given the set of extreme rays, the determination of the overall stoichiometry and subsequently the optimal yield is straight forward and only a minor computational or algorithmic task. One additional structure, the external stoichiometry matrix, is required though. It has the same structure

as the stoichiometry matrix we know from the use in previous chapters, also the reactions and thus the number of columns is the same. The only difference lies in the metabolites, which are disjoint, i.e. the metabolites of the environment marked as external are not used in the extreme ray enumeration.

Overall stoichiometry The overall stoichiometry is represented as a set of vectors. One vector for each extreme ray and each of the vectors has entries for the external metabolites. A positive entry is indicating that the respective metabolite is produced through the pathway and, vice versa, a metabolite is needed for consumption if it has a negative entry at its position.

To compute these vectors, we simply calculate for each entry of each vector the scalar product between the flux vector of the particular extreme ray and the row vector of the external stoichiometry matrix corresponding to the respective external metabolite.

$$v_{yield_{ij}} = v_{e_i} \bullet r_{m_j} \tag{36}$$

Optimal Yield The yield of a pathway is the ratio between the units of produced metabolites and the units of consumed metabolites. In this implementation, we also want to regard the goodness of a metabolite based on factors like its energy or graph indices. Therefore, when summing up the units of the metabolites they are multiplied by such a value indicating the goodness. So for each extreme ray we compute its yield from the corresponding vector of the stoichiometry in the following way:

$$yield_{e_i} = \frac{\sum \left(v_{yield_{ij}} > 0 \right)}{\sum \left(v_{yield_{ij}} < 0 \right)} \tag{37}$$

The optimal yield is now simply the maximal value among those, just being computed for all extreme rays. This value can now serve as a criteria for the comparison of different metabolic networks.

5.3 Experimental Study

In this section it is intended to show the effect of the network redundancies and the reaction order on the performance of the metabolic flux analysis tool. This is done on a data set that was used before by other authors for performance studies and comparisons. It is originally described by [KSGG03] for studies on the `FluxAnalyzer`. From the entire data set, see table 4, we will

	Glucose	**Acetate**	**Glycerol**	**Succinate**	**all 4 substrates simultaneously**
Participating reactions/metabolites	106/89	105/89	106/89	105/89	110/89
upper bound S_{max}	$3.69*10^{18}$	$5.57*10^{17}$	$3.69*10^{18}$	$5.57*10^{17}$	$4.39*10^{21}$
reduced network: participating reactions/metabolites	46/30	45/30	47/31	45/30	51/31
reduced network: upper bound S_{max}	$5.12*10^{11}$	$1.67*10^{11}$	$7.52*10^{11}$	$1.67*10^{11}$	$4.85*10^{13}$
computation time	1027 sec	3.8 sec	53.6 sec	16.4 sec	approx. 50 hours
number of EFMs (S)	27099	599	11332	4249	507632

Table 4: Data set [KSGG03]

only look at the acetate data because with 599 elementary modes it is the only one that we can seriously consider to use to check the correctness of our tool. As measure for the performance we will take some similar measure as used by [UW05b], but it should be noted that those results can not be compared with the results of this experimental study, as will be explained later in this section.

At first, we will observe the influence of the two different reduction steps on the computational cost of the algorithm and computation time of the flux analysis tool, respectively. In table 5 we have listed the total number of candidates which had to be generated and tested for adjacency in the course of one run with the acetate data. This might be the most important indicator for computational cost since it is known that the adjacency tests make up more than a half of the computational effort. Furthermore, the sum of all candidate flux modes that were successfully added in one of the iterations is listed. Since all candidates have to be tested against all current elementary modes, this number is also interesting for the estimation of the complexity of the algorithm. The rows and columns of the stoichiometry matrix are considered because they indicate the number of iterations and the number of binary OR-operations per adjacency test that have to be performed, respectively. Finally, with reference to the computation time, it can be studied which of the factors has the biggest impact on the performance. Now, all these measurements are taken for four different networks. That is, the original acetate network and three reduced networks of the same data. One only performing the reduction on the stoichiometry matrix S, another only the reduction on the kernel matrix K, and the last network performs both of the reduction steps. Inspecting the table, it is difficult to find one single factor which can account completely for the respective computational behaviors, thus those measurements thought to be

	No Reduction	Red. of S	Red. of K	Red. of S and K
Candidates	35491	13692	31322	24946
Candidates (added)	1898	1992	1534	1556
Rows(Metabolites)	88	39	61	30
Columns(Reactions)	104	55	77	46
Computation Time in s	27	11	21	13

Table 5: Performance measurements for different reductions

most important were combined and compared against the computation time again. As already explained, adjacency tests take up most of computation time and they can be estimated by the number of candidates times the size of the current set of elementary modes. Since we do not know the size of the sets for every iteration we take the number of added candidates as an approximation, which should be valid at least for the purpose of comparison. Adjacency tests in networks with many reactions are more expensive than those of smaller networks, since for each processed reaction an OR-operation has to be performed in every test. To simplify that, we just take the number of columns of the network. Consequently, our final combination is the product of all generated candidates, those added to the set of elementary modes and the number of columns. Figure 37 shows a histogram with this combination compared together with the computation time for all networks. As can be seen, the product correlates well with the computation time. Interestingly, in table 5 and 37, the number of candidates as well as the computation time of the fully reduced network is higher than for the network that did not perform the reduction of the stoichiometry matrix. It might also seem peculiar that the number of added candidates correlates rather little to the total number of candidates. Both observations can be explained by another observation which was mentioned in other experimental studies before and will be discussed in the next paragraph as well. It is the influence of the order of reactions in the kernel matrix. In general, it can be assumed that if reactions containing many zeros are processed first, then the number of adjacency tests is lower. In the case of the fully reduced network, the second reduction step might have brought the kernel matrix in a structure with fewer lines containing many zeros, but, on the other hand, reducing the total number of zeros in the entire matrix, resulting in the fewer added candidates.

Knowing the possible impact of the row ordering of the kernel matrix, it has been tried to use this knowledge to reduce the number of candidates which have to be generated to an optimum. A simple sorting of rows in K regarding the number of zeros they contain, starting with the maximal number, has in some cases reduced the number of candidates by an order of magnitude[UW05b]. In this chapter, we introduced an alternative method. Instead of sorting

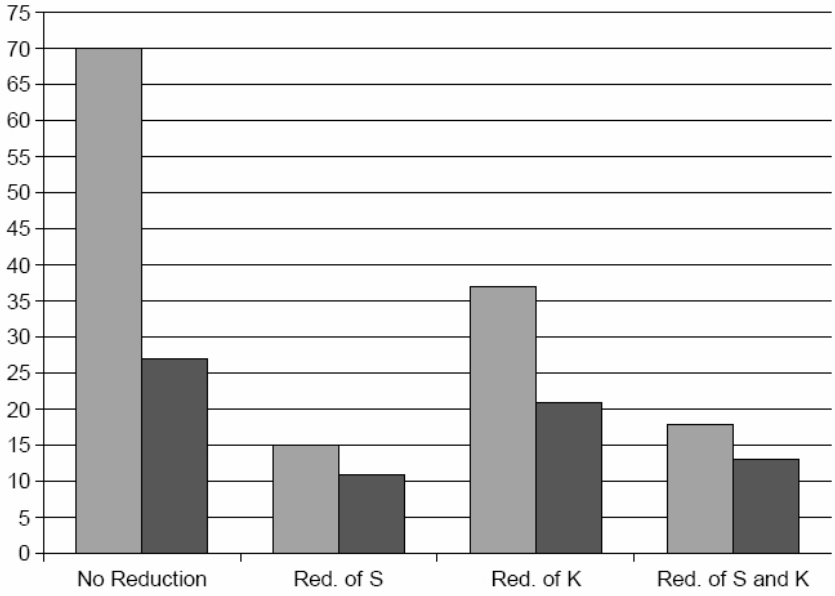


Figure 37: Comparison of reductions. Light gray = Candidates $\times$ Cols $\times$ addedCandidates in 10^8, Dark gray = Computation Time in s.

rows once before the first iteration, rows will be evaluated before each iteration and, thus, like a greedy algorithm, always take the row which is most likely to lead to a optimal solution. Since it is desired to reduce the number of adjacency tests which have to be performed between the candidates and the current elementary modes of each iteration, it seems more useful to take the number of candidates which will be generated by a row as the criteria for its goodness. In the null-space approach, this resembles the product of the number of fluxes with negative entries in the respective row with the number of positive fluxes. This product will be used instead of the number of zeros in a row. In figure 38 both methods are compared regarding the number of generated candidates and added candidates (both on log10 scale). Also, random row ordering is listed.

As already mentioned, a difference of an order of magnitude in the total number of candidates between a random row ordering and the descending order regarding the number of zeros can be observed. An almost similar drastic difference also exists between the initial sorting and our greedy-like approach of row ordering, suggesting the potential of this method to reduce computational cost to some extent. For this flux analysis tool and most likely for similar null-space approaches, our row-ordering seems to be the preferable option. However, no statements can be made about other approaches, as [UW05b], who use the initial sorting and provide similar measurements. Those results can not be compared since they use a null-space approach with re-

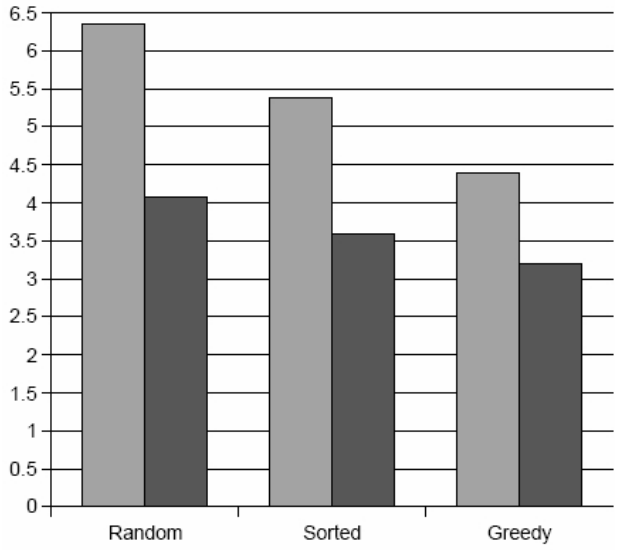


Figure 38: Comparison of different row orders. Light gray = generated candidates, Dark gray = added candidates. Number of candidates on log10 scale

versible reactions instead of exclusively irreversible ones. Therefore, generally expecting fewer but more expensive adjacency tests. Our observations still suggest that improvements may very well rest in this very simple but odd part of the algorithm.

6 Discussion

In this section, it is discussed whether the introduced simulation tool can provide the functionality as desired.

6.1 Results

A few simulation runs were performed and their results analyzed to gain information about the properties of the evolved metabolic networks. Discussed will be ten simulations. There are five different topological indices that can be used as selection criteria for reactions and metabolites. For each of them, two simulations were performed, where one is aiming to reduce the respective index and the other tries to maximize it. It is easier to observe different behaviors among the networks regarding only the different topological indices, thus energy calculation is not regarded in these simulations. All simulations are initialized with a population of six individuals. The genome for the individuals is chosen randomly, but for all simulations the random number generator (RNG [MN98]) used for building a random genome and generate random mutations is set with the seed number. The set of metabolites constituting the environment is the same in all simulations as well. Thus, the simulations start with equal preconditions. Furthermore, in every generation, half the population is selected and from each of the selected individuals a new individual is produced and a mutation in its genome is performed.

Metabolic networks are small world networks. Therefore, the metabolite connectivity distribution follows the power law. In other words, in a realistic metabolic network, a few highly connected metabolites, called hubs, should be observed, whereas the majority of metabolites is involved in only one or two reactions. In order to prove whether this property can be found in the networks produced by the simulation tool, the distribution of the metabolite connectivity was derived. The different simulations do not result in networks of equal size. It is also known that in small networks, around 50 metabolites, the connectivity distribution does not follow exactly the power law and contains fewer hubs than could expected in a scale free network. Consequently, the networks are grouped into sets of networks with similar size. The values of the connectivity distribution are listed in table 6 and illustrated in figure fig:expconnectivity.

In all networks, the majority of metabolites is involved in one or two reactions, but only larger networks ($m > 150$) contain enough highly connected metabolites to satisfy the small world property. A similar observation as in [PS05] can be made. In small networks, the number of hubs in the range between eight and twelve is higher than for scale-free networks, but too few hubs of higher connectivity exist. Most real world metabolic networks contain more than

$\frac{Connectivity}{\lvert Metabolites \rvert}$	1	2	3	4	5	6	7	8 - 12	>12
avg(m)=47.5	43.91	12.61	13.03	8.61	8.82	4.83	2.31	4.41	1.47
avg(m)=104.5	50.89	17.1	6.02	5.2	5.61	5.2	3.28	5.06	1.64
avg(m)=156.5	56.21	16.36	5.85	3.56	2.92	3.29	2.1	4.02	5.67
avg(m)=249.5	58.59	12.89	8.13	6.01	2.69	2	1.37	3.67	4.64
avg(m)=570	62.8	12.1	9.12	5.49	2.89	1.2	1.46	2.34	2.63

Table 6: Connectivity of metabolites in networks of different sizes. Frequency in %

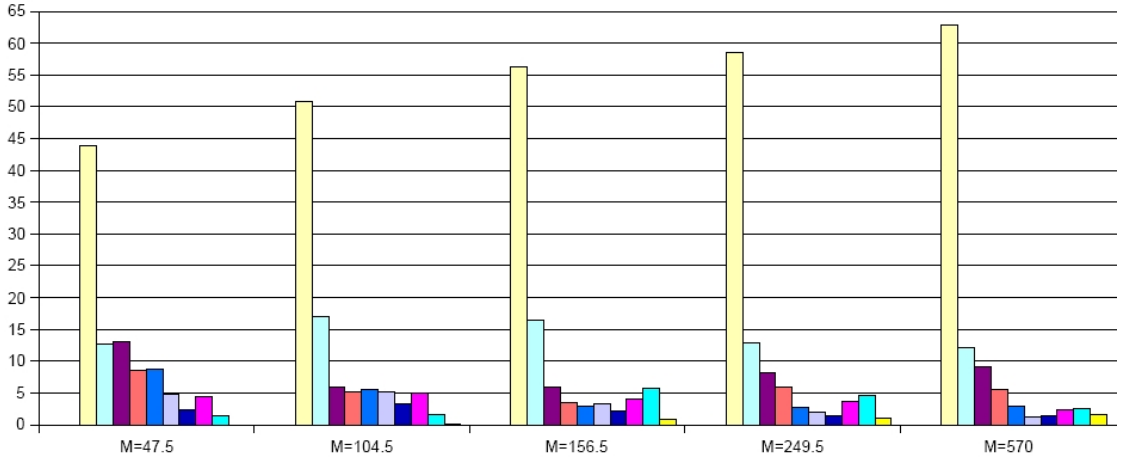


Figure 39: Connectivity of metabolites in networks of different sizes. Frequency in %. Connectivity of 1, 2, 3, 4, 5, 6, 7, 8-12, >12, >30 for all 5 network classes.

fifty metabolites, thus are large networks. The conclusion about the small-world property of metabolic networks, therefore, was drawn with the assumption that the network of investigation is large ($m \geq 100$). The connectivity distribution of smaller real world metabolic networks, actually, exhibit the same deviations from the power law as the networks gained from the example simulations. Accordingly, we can not state that all produced networks are scale-free, but we can assume them to resemble realistic metabolic networks.

For further analysis, one of the simulations is studied in more detail. On the example of the simulation that uses the minimal Balaban index as selection criteria, some of the information that can be used for a network analysis is illustrated. The part in the protocol representing the constructed network is given below for an individual of the first generation. In Figure 40 this network is illustrated as network graph. In Figure 41 and 42 follow network graphs for generation two and 87, respectively. Enzymes are drawn in light blue circles and metabolites in light gray boxes. The enzyme and metabolite indices are explained in the protocol as well, as is shown for one example each in the first generation.

```
m1 -> e45 in: 2
m1 -> e37 in: 4
m2 -> e12 in: 1
m4 -> e12 in: 0
m4 -> e37 in: 3
e12 -> m5 in: 0
e12 -> m6 in: 0
e12 -> m7 in: 1
e12 -> m8 in: 1
e45 -> m9 in: 2
e37 -> m10 in: 3
e37 -> m11 in: 4

C1=CCC=C1 m7

e45 = 412040
```

From the network graph in figure 42 it can be derived, that some enzymes catalyze many reactions (e2, e12, e44, e45) and others participate in very few reactions. In the different theories about the evolution of enzymes, it was stated that enzymes of low specificity evolve to

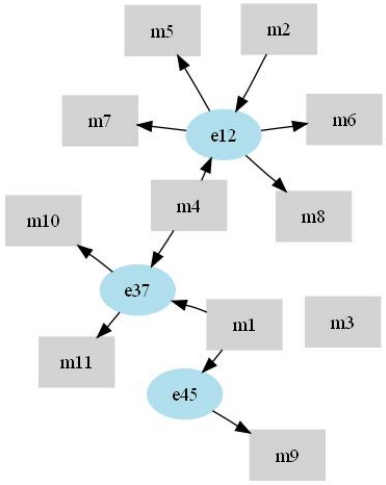


Figure 40: Example: Network graph from simulation Balaban in generation 1. Light bue circle = enzyme, light gray box = metabolite.

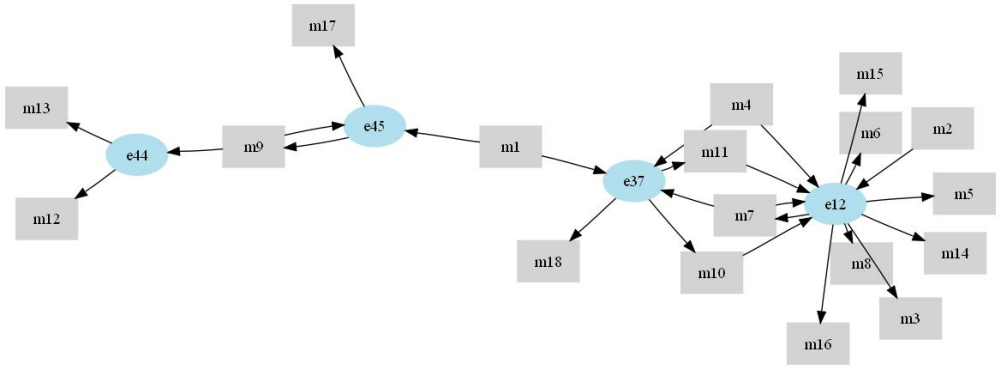


Figure 41: Example: Network graph from simulation Balaban in generation 2. Light bue circle = enzyme, light gray box = metabolite.

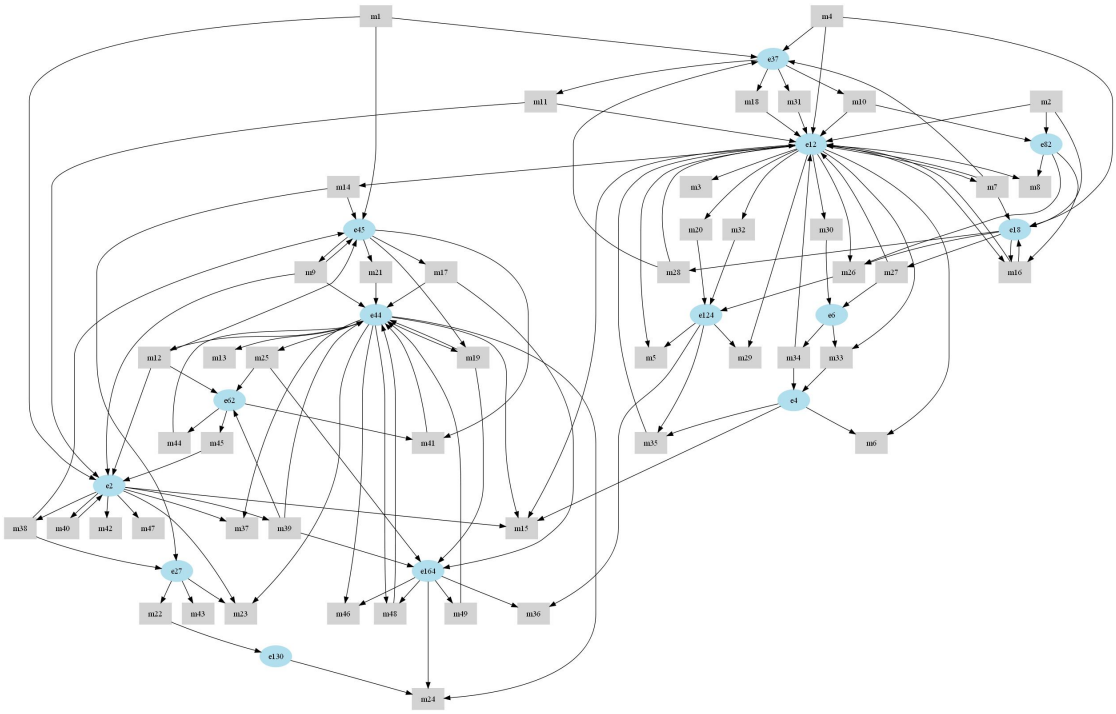


Figure 42: Example: Network graph from simulation Balaban in generation 87. Light bue circle = enzyme, light gray box = metabolite.

highly specific enzymes. This is expected for the simulations as well. To study the evolution of enzymes within a simulation run, a phylogenetic tree showing the evolutionary relationships among the enzymes can be constructed from the protocol information. The evolutionary tree for all enzymes in generation 87 is depicted in 43. Below is the protocol information for the first evolutionary tree.

```
130 = 412021 parents: 42414040 32412040 302040
90  = 302040 parents: 42414040 32412040
6  = 414041 parents: 42414040 32412040
51  = 32412040 parents: 42414040
164 = 20424020 parents: 42414040
21  = 42414040
```

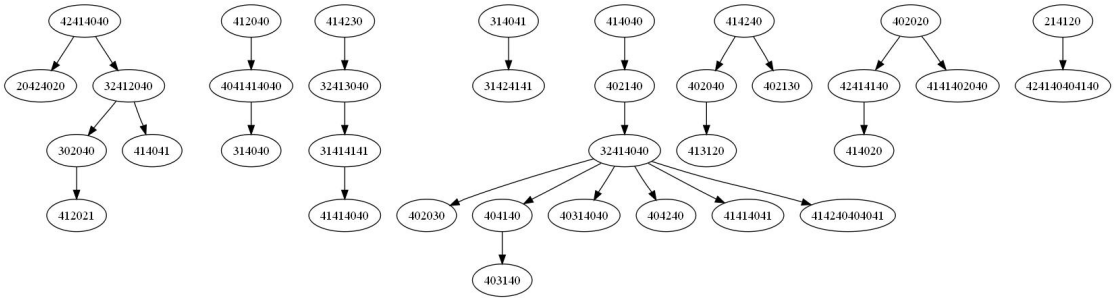


Figure 43: Example: Phylogenetic tree of reactions in simulation Balaban

More detailed information about the evolution of the enzymes can be derived by looking at the generation in which the respective enzyme participated for the first time. This information is listed in table 7 for all enzymes in the example network in generation 87. A tendency for early enzymes to be less specific can be observed, in fact, three of the four enzymes with relatively low specificity are from generation one or two. With the exception of $e2$, all enzymes in later generations are more specific. The same observation is made for the other simulations.

The observation stated above is explained with the following scenario. In the beginning, ev-

Enzyme	e37	e45	e12	e44	e18	e27	e6	e82	e4	e62	e124	e130	e2
Generation	1	1	1	2	3	3	7	10	14	42	44	51	53
Connectivity	4	5	12	9	4	2	2	2	2	2	3	1	6

Table 7: Specificity of enzymes in the example network

ery reaction producing valid metabolites is beneficial for the yield of the network. Since more metabolites are generated, more of the already existing enzymes find reactants. At some point, metabolites will protrude from the metabolite pool, that is some metabolites become more beneficial than others. This in turn means that not all of the enzymes increase the network yield. If an enzyme with low specificity overlaps in functionality with another enzyme which is specialized on the few common reactions, then the impact of the former enzyme on these reactions is very low. Since the rate of a reaction depends on how many reactions the involved enzyme performs, the remaining reactions of the lowly specific enzyme are performed on a, relatively, low rate. A highly specific enzyme which can catalyze the remaining non-overlapping reactions, can do so at much higher rate. Overall, it can be stated that enzymes which have a unique function have an advantage in natural selection. In later generations, most beneficial reactions are already realized by existing enzymes and only few are left. It follows that only specific enzymes can find their niche. However, sometimes lowly specific enzymes enter the network at later stages because they might express a completely new functionality, e.g. different atoms, bond type or reaction type. This scenario does not comply exactly with retrograde evolution[Hor45], since it does not need a metabolite depletion, but it integrates to a certain extent the idea of patchwork evolution[Jen76] and the theory of [KB84].

Further analysis could be done based on phylogenetic trees for evolutionary relationships among individuals. In figure 44 is an example tree for the simulation discussed above. At this place, there will be no detailed study in this direction. Investigations could be performed on individuals with many descendants (individual 48) and those child individuals which were the only one prevailing (individual 138). For this purpose, further advanced analysis of the network structure had to be performed. For example, metabolic flux analysis could be used to investigate the evolvability or the robustness against mutations of an individuals network.

6.2 Conclusion

The simulation model, presented in this thesis, can be a tool in the study of the evolution of metabolism and enzymes, as well as research on properties of complex networks. The underlying graph concept in combination with a sophisticated artificial chemistry ensures a realistic behavior of the evolution. The resulting metabolic networks exhibit the characteristic properties of real world metabolism. Various options, from the constitution of the environment and chemistry to selection properties, such as the number of descendants or the use of topological indices as additional criteria, can be adjusted. An extensive amount of information about the simulation can be gained from its protocol. The data about metabolic networks and their evolution over

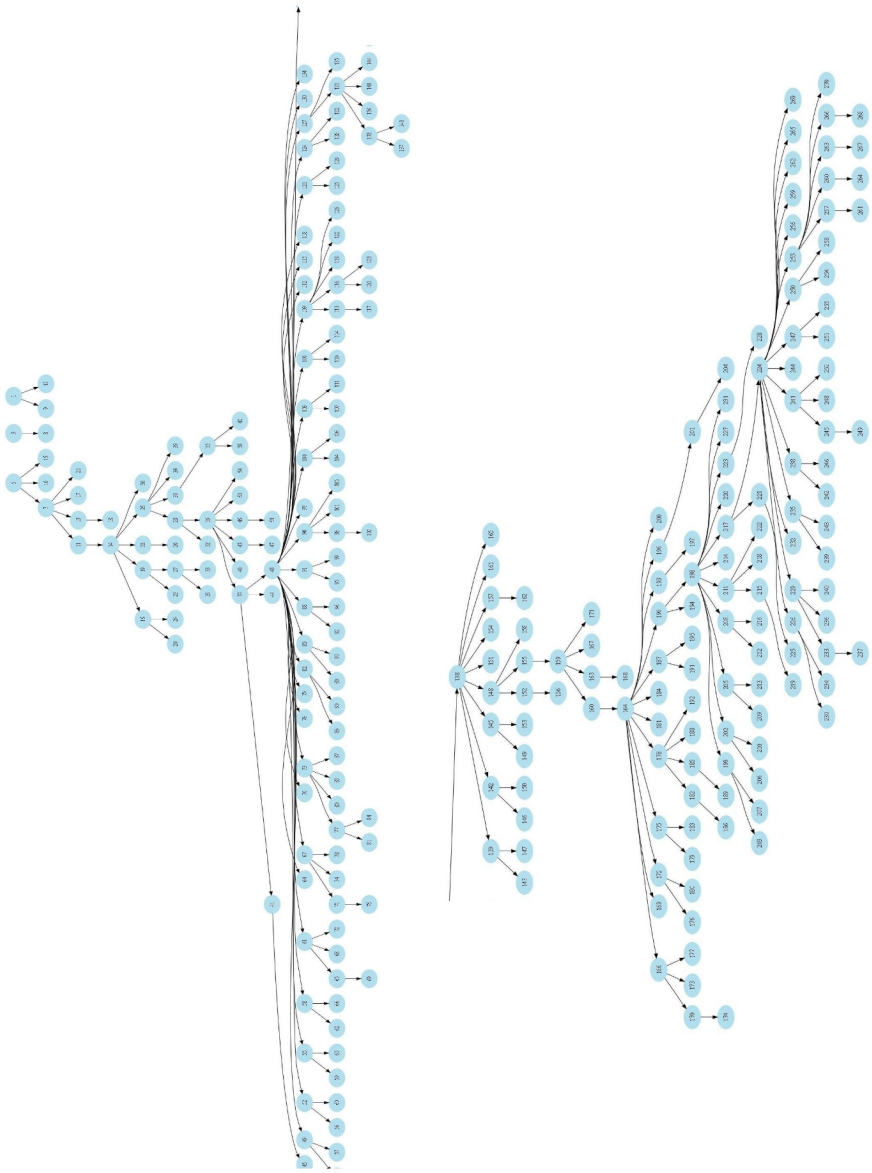


Figure 44: Example: Phylogenetic tree of individuals in simulation Balaban.

generations, is expressive and meaningful, so that it can be used to formulate new hypotheses or test existing theories.

6.3 Outlook

The presented work can be extended in several ways and on different levels. The first option would be to use the features of the integrated tools, `ToyChem` and metabolic flux analysis, more extensively. Besides the energy calculation, `ToyChem` also provides routines to calculate solvation energies and reaction rates. Consequently, a multi-phase environment, in which the individuals would evolve, can be modeled. The behavior of enzymes will resemble more realistic behavior, when considering reaction rates. From the metabolic flux analysis tool, which was also implemented in this work, so far only the yield determination has been used. Possible measures, that can be added to the assessment of networks, are robustness, flexibility or modularity. Therefore, extending the range of hypotheses which can be investigated. The second option for improvement is the implementation of new internal modules and extension of some of the existing concepts. One of the next projects will be the integration of regulatory elements, adding to the complexity of the simulated individuals, leading to a more realistic characteristic of the metabolism properties. Furthermore, the selection process can be object of change. At the moment the used selection procedure resembles an adaptive walk, but other effective evolutionary algorithms are known. The results of the existing adaptive walk, might also be improved by using other selection criteria, such as robustness, to account for the time-dependency of the fitness landscape. Additional changes could involve the metabolite pool. It is conceivable to integrate metabolite transporters and use absolute numbers of metabolites. Consequently, allowing depletion of metabolites and compartmentalization in an individual and regulating its excretion. Also the environment of an individual could depend on other individuals. A neighbor relationship could be introduced. Individuals would have to compete for metabolites in the environment and also process the excreted metabolites of neighbors. Besides all the changes of the model, a lot of the future work consists of developing ways to analyze the simulations and study the network properties in more detail. The focus will be on the emergence of robustness and flexibility.

References

[Are79] J.G. Arens. *Rec Trav Chim Pays-Bas*, 98:155–161, 1979.

[Bal82] A.T. Balaban. Highly discriminating distance-based topological index. *Chem Phys Lett*, 89:399–404, 1982.

[Ben02] G. Benkö. A toy model of chemical reaction networks. Master's thesis, Universität Wien, 2002.

[Ben06] G. Benkö. *Uncovering the Structure of an Artificial Chemistry Universe*. PhD thesis, Universität Leipzig, 2006.

[Ber67] J. Bernal. *The Origin of Life*. W. Clowes and Sons, London, 1967.

[BF03] G. Benkö and C. Flamm. A graph-based toy model of chemistry. *J Chem Inf Comput Sci*, 43(4):1085–1095, 2003.

[BF04] G. Benkö and C. Flamm. Multi-phase artificial chemistry. In *The Logic of Artificial Life: Abstracting and Synthesizing the Principles of Living Systems*, 2004.

[BF05a] G. Benkö and C. Flamm. Explicit collision simulation of chemical reactions in a graph based artificial chemistry. In *Proceedings of the 8th European Conference on Artificial Life (ECAL)*, 2005.

[BF05b] G. Benkö and C. Flamm. The toychem package: A computational toolkit implementing a realistic artificial chemistry model. In *Proceedings: Math/Chem/Comp 2005*, 2005.

[BU88] J. Bauer and I. Ugi. Interactive generation of organic reactions by igor2 and the pc-assisted discovery of a new reaction. *Tetrahedron Computer Methodology*, 1:129–132, 1988.

[CFSV04] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *Int J Pattern Recogn*, 2004.

[CS66] A.G. Cairns-Smith. The origin of life and the nature of the primitive gene. *J Theor Biol*, 10:53–88, 1966.

[CSFF05] L. Cabusora, E. Sutton, A. Fulmer, and C. V. Forst. Differential network expression during drug and stress response. *Bioinformatics*, 21(12):2898–2905, 2005.

[Dar93] C. Darwin. *The origin of species*. Random House, 1993.

[Day76] M. Dayhoff. The origin and evolution of protein superfamilies. *Fcd Proc*, 35:2132–2138, 1976.

[DMPRS07] J. J. Diaz-Mejia, E. Perez-Rueda, and L. Segovia. A network perspective on the evolution of metabolism by gene duplication. *Genome Biol*, 8(2), 2007.

[DO80] D. Deamer and J. Oro. Role of lipids in prebiotic structures. *BioSystems*, 12:167–175, 1980.

[DU73] J. Dugundji and I. Ugi. An algebraic model of constitutional chemistry as a basis for chemical computer programs, 1973.

[Eak63] R. Eakin. An approach to the evolution of metabolism. *Proc Natl Acad Sci U S A*, 49(3):360–6, 1963.

[ES79] M. Eigen and P. Schuster. *The Hypercycle, a Principle of Natural Self-Organization*. Springer-Verlag, 1979.

[F.D86] F.DeTar. Modern approaches to chemical reaction searching. *Comput Chem*, 11:227, 1986.

[Fuj86a] S. Fujita. Description of organic reactions based on imaginary transition structures. 1. introduction of new concepts. *J Chem Inf Comput Sci*, 26(4):205–212, 1986.

[Fuj86b] S. Fujita. Description of organic reactions based on imaginary transition structures. 2. classification of one-string reactions having an even-membered cyclic reaction graph. *J Chem Inf Comput Sci*, 26(4):212–223, 1986.

[Fuj86c] S. Fujita. Description of organic reactions based on imaginary transition structures. 3. classification of one-string reactns having an odd-membered cyclic reaction graph. *J Chem Inf Comput Sci*, 26(4):224–230, 1986.

[Fuj86d] S. Fujita. Description of organic reactions based on imaginary transition structures. 4. three-nodal and four-nodal subgraphs for a systematic characterization of reaction. *J Chem Inf Comput Sci*, 26(4):231–237, 1986.

[Fuj86e] S. Fujita. Description of organic reactions based on imaginary transition structures. 5. recombination of reaction strings in a synthesis space and its application to the description of synthetic pathways. *J Chem Inf Comput Sci*, 26(4):238–242, 1986.

[Fuj87a] S. Fujita. Description of organic reactions based on imaginary transition structures. 6. classification and enumeration of two-string reactions with one common node. *J Chem Inf Comput Sci*, 27(3):99–104, 1987.

[Fuj87b] S. Fujita. Description of organic reactions based on imaginary transitions structures. 7. classification and enumeration of two-string reactions with two or more common nodes. *J Chem Inf Comput Sci*, 27(3):104–110, 1987.

[Fuj87c] S. Fujita. Structurereaction type paradigm in the conventional methods of describing organic reactions and the concept of imaginary transitions structures overcoming this paradigm. *J Chem Inf Comput Sci*, 27(3):120–126, 1987.

[Gas03] J. Gasteiger, editor. *Handbook of Chemoinformatics: From Data to Knowledge*, volume 1. WILEY, August 2003.

[GK04] J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5(175), November 2004.

[Har75] H. Hartman. Speculations on the origin and evolution of metabolism. *J Mol Evol*, 4:359–370, Mar 1975.

[Hen74] J. Hendrickson. The variety of thermal pericyclic reactions. *Angew Chem internat Edit*, 13:47–76, 1974.

[Hen79] J. Hendrickson. A systematic organization of synthetic reactions. *J Chem Inf Comput Sci*, 19(3):129–136, 1979.

[Hen97] J. Hendrickson. Comprehensive system for classification and nomenclature of organic reactions. *J Chem Inf Comput Sci*, 37(5):852–860, 1997.

[Her90] R. Herges. Reaction planning: prediction of new organic reactions. *J Chem Inf Comput Sci*, 30(4):377–383, 1990.

[Him] M. Himsolt. *GML: A portable Graph File Format*. Universität Passau.

[Hor45] N. Horowitz. On the evolution of biochemical syntheses. *Proc Nat Acad Sci*, 31:153–157, 1945.

[Jen76] R. Jensen. Enzyme recruitment in evolution of new functions. *Ann Rev Microbiol*, 30:409–425, 1976.

[Kau93] S. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, May 1993.

[KB84] H. Kacser and R. Beeby. Evolution of catalytic proteins. *J Mol Evol*, 20:38–51, 1984.

[KH76] L.B. Krier and L.H. Hall. Molecular connectivity in chemistry and drug research. *Academic Press*, 1976.

[KH04] E. B. Krissinel and K. Henrick. Common subgraph isomorphism detection by backtracking search. *Software Practice and Experience*, 34:591–607, 2004.

[KM94] P. D. Karp and M. L. Mavrovouniotis. Representing, analyzing, and synthesizing biochemical pathways. *IEEE Expert*, 9(2):11–21, 1994.

[KSGG03] S. Klamt, J. Stelling, M. Ginkel, and E.D. Gilles. Fluxanalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. *Bioinformatics*, 19.2:261–269, 2003.

[Lia96] J. Liao. Pathway analysis, engineering, and physiological considerations for redirecting central metabolism. *Biotechnol Bioeng*, 52:129–140, 1996.

[LK97] A. Lyubarev and B. Kurganov. Origin of biochemical organization. *BioSystems*, 42:103–110, 1997.

[Mil53] S. Miller. A Production of Amino Acids Under Possible Primitive Earth Conditions. *Science*, 117(3046):528–529, 1953.

[MN98] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans Model Comput Simul*, 8(1):3–30, 1998.

[Mot53] T.S. Motzkin. The double description method. *Ann Math*, 8:51–73, 1953.

[Nus78] M.D. Nussinov. Formation of the early earth regolith. *Nature*, 275:19–21, 1978.

[Pal06] B. O. Palsson. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, New York, NY, USA, 2006.

[Pla47] J.R. Platt. Influence of neighbor bonds on additive bond properties in paraffins. *J Chem Phys*, 15:419–420, 1947.

[PNW+03] J. Papin, N.Price, S. Wiback, D. Fell, and B.O.Palsson. Metabolic pathways in the post-genome era. *Trends Biochem Sci*, 28(5):250–258, May 2003.

[PS05] T. Pfeiffer and O. Soyer. The evolution of connectivity in metabolic networks. *PLoS Biology*, 3/7:228, 2005.

[PSVN+99] T. Pfeiffer, I. Sanchez-Valdenabro, J.C. Nuno, F. Montero, and S. Schuster. Metatool: for studying metabolic networks. *Bioinformatics*, 15.3:251–257, 1999.

[Ran75] M. Randic. Characterization of molecular branching. *J Am Chem Soc*, 97(23):6609–6615, 1975.

[RD94] M.J. Russel and R.M. Daniel. A hydrothermally precipitated catalytic iron sulphide membrane as a first step toward life. *J Mol Evol*, 39:231–243, 1994.

[Sch99] S. Schuster. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol*, 17:53–60, 1999.

[SHWF02] S. Schuster, C. Hilgetag, J.H. Woods, and D.A. Fell. Reaction routes in biochemical reaction systems: Algebraic properties, validated calculation procedure and example from nucleotide metabolism. *J Math Biol*, 45:153–181, 2002.

[Sin] S. Singh. A universal power law and proportionate change process characterize the evolution of metabolic networks.

[SPM+02] S. Schuster, T. Pfeiffer, F. Moldenhauer, I. Koch, and T. Dankedar. Exploring the pathway structure of metabolism: decomposition into subnetworks and application to mycoplasma pneumoniae. *Bioinformatics*, 18.2:351–361, 2002.

[SS91] S. Schuster and R. Schuster. Detecting strictly detailed balanced subnetworks in open chemical reaction networks. *J Math Chem*, 6:17–40, 1991.

[SS06] A. Samal and S. Singh. Low degree metabolites explain essential reactions and enhance modularity in biological networks. *BMC Bioinformatics*, 7:118, 2006.

[SSF] T. Dandekar S. Schuster and D. Fell. Description of the algorithm for computing elementary flux modes.

[Tri92] N. Trinajstic. *Chemical Graph Theory, Second Edition (New Directions in Civil Engineering)*. CRC, February 1992.

[UW05a] R. Urbanczik and C. Wagner. Functional stoichiometric analysis of metabolic networks. *Bioinformatics*, 21.22:4176–4180, 2005.

[UW05b] R. Urbanczik and C. Wagner. An improved algorithm for stoichiometric network analysis: theory and applications. *Bioinformatics*, 21.7:1203–1210, 2005.

[Wäc88] G. Wächtershäuser. Before enzyme and templates: theory of surface metabolism. *Microbiol Rev*, 52:452–484, 1988.

[Wag04] C. Wagner. Nullspace approach to determine the elementary modes of chemical reaction systems. *J Phys Chem B*, 108:2425–2431, 2004.

[Wei88] D. Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J Chem Inf Comput Sci*, 28(1):31–36, 1988.

[Wei89] D. Weininger. Smiles. 2. algorithm for generation of unique smiles notation. *J Chem Inf Comput Sci*, 29(2):97–101, 1989.

[WF01] A. Wagner and D.A. Fell. The small world inside large metabolic networks. *Proc R Soc Lond B Biol Sci*, 268:1803–1810, 2001.

[Wie47] H. Wiener. Structural determination of paraffin boiling points. *J Am Chem Soc*, 69(1):17–20, 1947.

[Wie02] W. Wiechert. Modeling and simulation: tools for metabolic engineering. *J Biotechnol*, 94:37–63, 2002.

[ZBP94] N.S. Zefirov, I.I. Baskin, and V.A. Palyulin. Symbeq program and its application in computer-assisted reaction design. *J Chem Inf Comput Sci*, 34(4):994–999, 1994.

I affirm, that I have written this work independently and only used the sources and resources listed in the bibliography.

Leipzig, February 1, 2008 Alexander Ullrich

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig, February 1, 2008 Alexander Ullrich