universität wien

MASTERARBEIT

Titel der Masterarbeit

„Enzyme Mechanisms as Synthesis Planning Problem“

verfasst von

Bernhard Thiel, BSc

angestrebter akademischer Grad

Master of Science (MSc)

Wien, 2013

Studienkennzahl lt. Studienblatt:	A 066 862
Studienrichtung lt. Studienblatt:	Masterstudium Chemie
Betreut von:	ao. Univ.-Prof. Mag. Dr. Christoph Flamm

Contents

Zusammenfassung

Bei der Untersuchung und Analyse von metabolischen Netzwerken ist es oft notwendig zu wissen, welche Atome aus den Edukten bei einer chemischen, im biochemischen Kontext speziell einer enzymatischen, Reaktion zu welchen Atomen aus den Produkten werden. Leider fehlt diese Information in vielen Datenbanken. Da sich diese sogenannte Atom Map aus dem Reaktionsmechanismus ergibt, wäre also ein Programm, welches einen Reaktionsmechanismus für eine gegebene enzymatische Gesamtreaktion vorschlägt, von großem Nutzen.

Ferner könnte ein solches Programm auch neue Reaktionen, welche in der Natur noch nicht nachgewiesen wurden, aber theoretisch als enzymatische Reaktion denkbar wären, vorhersagen.

Ziel dieser Masterarbeit war es, ein solches Programm zu schreiben, welches dieses Problem als Syntheseplanungsproblem auffasst. Dabei wurde eine Breitensuche implementiert, die auf das Startmaterial, bestehend aus Substratmolekülen und katalytischen Gruppen im Enzym, alle möglichen chemischen Reaktionen anwendet und dadurch neue potentielle Enzym-Zustände generiert. Im nächsten Schritt werden nun auf jeden dieser neu generierten Zustände erneut alle Reaktionen angewendet. Dies geschieht in einer bidirektionalen Suche, das heißt, es wird sowohl von den Edukten aus Richtung Produkte gesucht, als auch von den Produkten Richtung Edukte. Dabei muss sichergestellt werden, dass ein möglicher Reaktionsmechanismus alle chemischen Gruppen, welche Teil des Enzyms sind, am Ende des Reaktionspfades wieder in den selben Zustand bringt, den sie am Anfang hatten. Diese Forderung schränkt die Möglichkeiten der Syntheseplanung weiter ein. Da die Zahl der generierten Zustände auf diese Weise mit der Suchtiefe sehr rasch exponentiell wächst, wurde für größere Suchtiefen eine Heuristik entwickelt.

Indem mitgezählt wurde, wie viele Stück von welchem Molekül im jeweiligen Zustand des Enzyms vorhanden sind, konnte die Dauer der Suche bereits drastisch reduziert werden. Durch die implementierten Heuristiken kann ein weiterer Geschwindigkeitsgewinn erziehlt werden, jedoch auf Kosten der Qualität der Ergebnisse. Insbesondere besteht die Gefahr, dass korrekte Pfade nicht gefunden werden, wenn die Balance zwischen Vollständigkeit und Geschwindigkeit bei der Heuristik zu sehr auf Seiten der Geschwindigkeit liegt.

Außerdem wurden über 700 Reaktionsregeln als Graphgrammatikregeln für Elementarreaktionen aus einer Reaktionsdatanbank extrahiert und sorgfältig aufbereitet, um für das Programm als Wissensgrundlage zu dienen.

Schließlich wurde für einige Beispiele aus einer Datenbank von Enzymen, für welche keine Gen-Sequenz bekannt ist, ein Reaktionsmechanismus vorgeschlagen. Unter anderem wurde der klassische Aldehyd-Dehydrogenase-Mechanismus vorgeschlagen, obwohl keine Regeln von einer Aldehyd-dehydrogenase dem Programm übergeben wurden. Dies zeigt sehr schön, dass durch Kombination von Elementarreaktionan aus verschiedenen Enzymen chemisch korrekte neue Mechanismen generiert werden können.

Abstract

When metabolic networks are examined and analyzed, it is often necessary to know which atoms of the educts correspond to which atoms of the products of a chemical reaction. In biochemical context enzymatic reactions are especially of interest. Unfortunatly this information is missing in many databases. Since this so called atom map can be derived from the reaction mechanism, it would be useful to have a program that can propose a mechanism for a given enzymatic reaction.

Furthermore such a program could also predict new reactions that have not yet been observed in nature but would be plausible enzymatic reactions.

The goal of this thesis was to write such a program, which treats the problems of finding the reaction mechanism as a synthesis planning problem. To this end a breadth first search was implemented which applies all possible reaction rules to the starting materials. These starting materials consist of the substrates and catalytic groups found in the enzyme. Application of these rules yields all possible new states of the enzyme. In the next steps all rules are applied to all these newly generated states. The whole procedure is done in a bidirectional search starting both from the educts and from the products. A possible reaction mechanism must ensure that all chemical groups provided by the enzyme are restored to the original state at the end of the reaction path. This property puts further restrictions on the synthesis planning. As the number of states that are thus generated grows exponentially, a heuristic was implemented for larger search depths.

By keeping count of the number of instances of each molecule at any state of the enzyme, the duration of the search could be dramatically reduced. The implemented heuristics can be used to reduce search time further, but this reduces the quality of the predictions. If the user puts the balance between speed and exhaustiveness too much to the side of speed, there is the danger that correct paths will not be found.

Furthermore over 700 graph grammar rules for elementary reactions were extracted from a reaction database and prepared for the use as a knowledgebase in this program.

Finally the program was applied to some examples from a database of orphan enzymes. Indeed a reaction mechanism could be proposed for these reactions. In one example the typical mechanism for an aldehyde dehydrogenase was proposed, altough no reaction rules for aldehyde dehydrogenase were given to the program. This shows how it is possible to generate new, chemically correct mechanisms by combination of elemenary reactions from different enzymes.

Part I

Introduction

Chapter 1

Cheminformatics

1.1 Synthesis planning programs

The first pioneer who had the vision of a computer system that would be able to classify chemical reactions and discover synthetic routes was Vladutz.[1]

In 1967 E.J.Corey formulated general principles of the design of synthetic routes to a given target and made a first attempt at the "detailed definition of the elements of Synthesis and their mutual interaction, in a most general sense"[2]. He formalized the idea of retrosynthesis and laid out the most important concepts and axioms of modern organic synthetic chemistry. Corey was rewarded with the Nobel Prize for the "development of the theory and methodology of organic synthesis".

These principles correspond to the view of synthesis planning as a search tree: The target molecule corresponds to the root of the tree, possible intermediates correspond to the nodes (with leaf nodes corresponding to available starting materials) and edges correspond to retrosynthetic transforms. The pathway from the root to a leaf of the tree corresponds to a retrosynthetic route.[3]

It was Corey and his group who, beside wet-chemical work, used these ideas to develop the first synthesis planning program LHASA.

In this section I will give a short overview over some of the most important synthesis planning programs.[4–6]

1.1.1 LHASA - Logic and Heuristics Applied to Synthetic Analysis

The LHASA program[7] was developed and extended over a period of over 25 years and has an extensive published history with contributions in many aspects of cheminformatics.[4]

Internally, molecules are coded as connection tables (*vide infra*). A sophisticated perception module is used to identify rings, functional groups, etc.[8]

The LHASA program uses a new language, CHMTRN, to code the chemical knowledge about reactions. LHASA's approach to chemical reaction rules is based on functional groups and combinations of functional groups that allow for certain reactivities.[9]

LHASA builds a search tree by applying retrosynthetic reaction rules to the starting material and thus generating new molecules. By iterative application of the retrosynthetic reaction rules to the molecules that were generated in the previous step, a search tree is built. Rules are clustered into different types (e.g. functional group exchange of functional group addition). Different types of rules are allowed at different stages of the search.

To avoid combinatorial explosion the program uses a semi-interactive approach. The user can choose among 5 major synthesis strategies.[10] Depending on the strategy selected, only certain rules are applied and some parts of the tree are early pruned. Furthermore the user can choose to further explore some intermediates and to prune the tree at others or to change strategy at some point of the tree. These strategies are the short-range strategy (also referred to as functional group based strategy), the topological strategy, the long-ranged (or transform based) strategy, the stereochemical strategy and the starting material strategy.

The short range strategy uses the present functional groups to achieve strong simplifications in the target molecule. Remember that LHASA searches

in retrosynthetic direction from the target to the starting materials. Thus simplification of the target molecule is usually desired.

The topological strategy uses strategic bonds[11] that the program will try to break in order to generate smaller and ideally simpler building blocks. Strategic bonds can be identified by the user or by the program. In order to break the strategic bond which leads to a lot of simplification, some steps that make the target more complex might be needed beforehand. It might be, for example, necessary to introduce a certain functional group before a rule can be applied that breaks the strategic bond. LHASA can identify those needed functional groups and introduce them.

The transform based strategy is centered around a certain reaction rule (e.g. the Robinson annulation). LHASA will try to modify the target molecule to prepare it for the application of the chosen transform.

The stereochemical strategy tries to find retrosynthetic transforms to stereochemical simpler precursors. Ideally these transforms should generate the desired stereochemistry in a high diastereomeric access when applied in the forward direction.

The best retrosynthetic path for real word applications is not always the one that achieves as much simplification of the target molecule as possible, but might be the one that transforms the target molecule into cheap, readily available but probably rather complex starting materials, like natural products. The starting material strategy of LHASA[12] lets the user draw an appropriate starting material or choose from a pool of suggestions generated by LHASA. Then the program identifies the best mapping of the starting material onto the target. After that the different retrosynthetic goals are identified, as there might be several positions where the starting material has to be modified, and a priority is assigned to them. Retrosynthetic search is then performed to achieve those goals one after the other.

Additionally the LHASA program has a module that identifies which functional groups need protection in a certain step and suggests protective groups.[13]

1.1.2 SYNCHEM

The SYNCHEM program uses a heuristic best first search approach for synthesis planning.[14] In their model the search tree is a bipartite graph consisting of compound and subgoal (or reaction) nodes. A compound node is solved if it is an available starting material or if any one of its children is solved. This corresponds to the fact that one way of synthesizing a compound is enough to make this compound synthesizable. Thus compound nodes are OR-nodes. A subgoal node, on the other hand, is only solved if all its children are solved (AND node), because all reagents are necessary to allow a chemical reaction.

Each node gets a merit that is 100 for starting material compounds and recursively depends on the children of the node for all other nodes. For OR nodes it depends on the best child only, for AND nodes it depends on all children. Of course the exact merit of a node can only be calculated if all its child nodes are solved, which means that also grand children and the grand-grand children etc. have to be solved. Thus the exact merit could only be used once the optimal solution is found and no heuristic is needed any more. To apply the merit for the heuristic, SYNCHEM thus uses an estimation function to estimate the merits of unexplored nodes.

In the best-first search the current best pathway is selected for further exploration. All possible reaction rules are then applied to the compound with the lowest merit in that pathway and new subgoal nodes are generated together with their children. The merit of newly generated compounds is estimated and used to calculate the merit of the subgoal nodes and, moving up the tree, update their precursors. Then the new best path is chosen and further explored.

A later development of SYNCHEM was a machine learning approach that allowed the program to extract the knowledge base of possible chemical reactions from databases.[15]

1.1.3 Route Designer

Route designer[16] is a relatively new synthesis planning program that has used extraction of reaction rules from databases right from the beginning.

Reaction rules are generated from the reactions found in databases in five steps[16]: First, all atoms that change their bonds during the reaction are identified. They form the reaction core. Then all relevant neighboring atoms are added to the reaction core to form the extended reaction core (functional group completion). In a third step the reaction cores are clustered into common groups. From these clusters a generalized rule template is generated which is then converted into a complete reaction rule.

To reduce the combinatorial explosion, Route designer uses some heuristics and allows the user to specify bonds that have to be broken and bonds that are unbreakable. The reactions are classified into several categories of which only regular disconnective transforms (that simplify the carbon skeleton in retrosynthetic direction) are always allowed. After all allowed transforms have been applied to a molecule in the search queue, these new transforms are examined and only the children of the most important transforms are queued for further exploration while all the others are stored as terminating nodes. For this purpose the transforms applied to a molecule are ordered by several criteria, among them the simplification of the target, the wastage of heavy atoms that go into side products of the reaction and the number of precedent reactions in the database.

1.1.4 Formal approaches - IGOR, EROS, SYNGEN

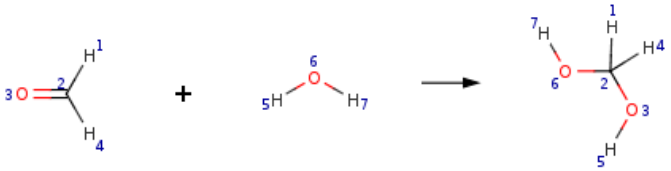
The programs described above use a knowledge base of known or good reactions. Therefor they only apply reactions that are known to work for similar molecules, which makes the synthetic paths they suggest more probable to actually work in laboratory and which allows for certain types of heuristics. However, the quality of their predictions strongly depends on the quality of the knowledge base which is extracted from databases for some approaches and handwritten in a cumbersome procedure for other approaches. Especially in the later case this knowledge base is automatically limited and can only cover a small part of all known chemistry.

Formal approaches on the other hand use a formal definition of chemical reactions. They only use general principles and do not depend on a knowledge base. They rely on a well-defined mathematical basis and try to avoid empirical rules as much as possible. They can suggest reactions that are unprecedented in literature, a fact that can be seen as an advantage or disadvantage depending on

the preferences of the user. While these approaches remove the potential bias of user-guided search strategies, the problem of the combinatorial explosion is even more challenging.[4]

Programs that use a formal approach are IGOR and RAIN[17], EROS[18,19] and SYNGEN[20,21]. IGOR and RAIN use the Dugundji-Ugi model (DU-model) of chemical reactions.

The DU-model



$$\begin{pmatrix} & H^1 & C^2 & O^3 & H^4 & H^5 & O^6 & H^7 \\ H^1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ C^2 & 1 & 0 & 2 & 1 & 0 & 0 & 0 \\ O^3 & 0 & 2 & 4 & 0 & 0 & 0 & 0 \\ H^4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ H^5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ O^6 & 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ H^7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} & H^1 & C^2 & O^3 & H^4 & H^5 & O^6 & H^7 \\ H^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C^2 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ O^3 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ H^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ H^5 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ O^6 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ H^7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} & H^1 & C^2 & O^3 & H^4 & H^5 & O^6 & H^7 \\ H^1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ C^2 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ O^3 & 0 & 1 & 4 & 0 & 1 & 0 & 0 \\ H^4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ H^5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ O^6 & 0 & 1 & 0 & 0 & 0 & 4 & 1 \\ H^7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 1.1: A chemical reaction in structural formula representation and in the DU model.

The DU model[17] uses a $n \times n$ matrix to represent a molecule or an ensemble of molecules (EM). The i^{th} atom of the EM corresponds to the i^{th} line and the i^{th} column of the matrix. The diagonal elements give the number of non-bonding valence electrons while the off-diagonal elements give the order of the bond between the two atoms. Chemical reactions are represented as reaction matrices that encode the change in electrons. By addition of the reaction matrix to the molecule matrix one gets a new molecule matrix.

An extension to the DU model was developed for aromatic systems and multi center bonds which cannot be represented in the classical DU-model.

Stereochemical information, on the other hand, is separated from the constitutional information on purpose. Ugi, Dugundji *et al.* developed the theory of the chemical identity group (CIG) to formalize stereochemistry.[17]

IGOR and RAIN

IGOR and RAIN are multi-purpose programs that can be used for synthesis planning among other applications.[17] They can solve the reaction equation $\boldsymbol{B} + \boldsymbol{R} = \boldsymbol{E}$, where $\boldsymbol{B}$ and $\boldsymbol{E}$ stand for EMs and $\boldsymbol{R}$ is the reaction matrix. IGOR solves the equation for a given reaction pattern $\boldsymbol{R}$, while RAIN solves it for a given educt EM $\boldsymbol{B}$. Thus IGOR can be used to invent new reactions that correspond to a given pattern of electronic change. RAIN is capable of

mono- and bidirectional searches for a reaction path between two given EMs. Geometric considerations can be used to guide the search towards a short path between the two EMs.[17]

Some additional programs that accomplish several tasks go together with IGOR and RAIN: PEMCD can be used to find the atom mapping between two EMs that corresponds to the minimal chemical distance[22] (the minimal redistribution of electrons). CANON[23] is used to derive a canonical numbering of the atoms in EMs. CORREL-S is used to correlate a target molecule with substructures that can serve as potential starting material. If the reaction of the starting materials (calculated with CORREL-S or given by the user) to the desired target is not balanced, STOECH[24] is used to calculate co-products in order to balance the reaction.

EROS

EROS is a semi-empirical program. It uses a formal treatment of reactions as a core, but is extended with a knowledge base that is well separated from the core program and can thus be extended or changed easily.[18,19]

To allow for the treatment of all sort of chemistry (not only wet-chemical synthesis in a flask), the concepts of reactors, phases, modes and kinetics were implemented. The mode of a reaction can be set to "monomolecular" (including pseudo-monomolecular reactions with the solvent), "interface" (only reactions between molecules of different phases), "mix" (every substrate can react with every substrate) or "mix_no_A+A" (like "mix", but two instances of the same substrate can not react with each other).

In the rules stored in the separate knowledge base, these concepts can be used to define heuristic rules which limit the search space. Furthermore EROS uses the subsystem PETRA[25] to calculate estimated values for different physicochemical properties of molecules. These properties can also be used by the rules in the knowledge base.

The developers of EROS also worked on a way to extract the chemical knowledge from reaction databases.[26] Therefore they used a Kohonen map approach.

SYNGEN

The program SYNGEN[20,21] separates the problem of synthesis planning into two parts: the retrosynthetical dissection of the target molecule's carbon skeleton or backbone and the generation of the functionality necessary to assemble these fragments to the target molecule in synthetic direction. The SYNGEN program searches for an ideal synthesis in the sense that it should be as short as possible and it should be convergent. Retrosynthetically a convergent synthesis can be achieved by cutting the target molecule into two halves and then cutting each half again into two pieces.

SYNGEN thus searches for an ideal way to break up the target's skeleton into pieces which can be found as skeletons of available starting materials in a way that corresponds to a convergent synthesis. Hendrikson uses the term "ordered bondset" to describe the set of bonds that should be broken in retrosynthetic direction together with the order in which they have to be broken.

SYNGEN then examines the changes in functionality that occur when the pieces are coupled and calculates the types of functionality required on the starting materials. The program only accepts a route if all necessary starting materials can be found in the database of available starting materials.

Chemical reactions are treated in a very formal way and functionality on carbons is classified by two numbers. All reactions that can form skeletal bonds are generated from three half-reactions.

1.1.5 Forward approaches - SYNOPSIS

Some programs do not search retrosynthetically from the target molecule but search in forward direction from the starting material. As an example of this forward approach, I will describe the program SYNOPSIS[27] in this section.

SYNOPSIS was primarily designed to find new candidate molecules for drug development. Thus the aim of the program is not to find a synthetic route to a specified target, but to find new, easily synthesizable candidate molecules that have certain properties and to propose a synthetic route towards them. Together with a starting material database and the set of allowed reactions, a fitness function is needed that depends on the desired property.

SYNOPSIS picks a random starting material out of a database and randomly chooses one reaction that can be applied to this molecule. Application of reactions is based on a functional group approach. A new molecule is generated and its fitness is calculated. Then this molecule is added to the starting materials and the next iteration begins. As the procedure continues, the selection of educts for new reactions is more and more driven towards molecules with a high fitness value and it becomes less likely for molecules with a fitness value lower than the currently best to become selected.

For drug discovery application, force field methods can be used to calculate the fitness of the molecules. In that case this is the computationally most expensive part of the algorithm.

At the end SYNOPSIS proposes new molecules with a high fitness value and a synthetic route towards them to the user.

1.1.6 Search for metabolic pathways

Félix *et al.*[28] describe an approach that uses artificial chemistry for the reconstruction of metabolic pathways. They move to the next level of abstraction compared to normal synthesis planning programs. While synthesis planning programs are interested in the generation of a single molecule, Félix *et al.* are interested in multi-molecules, that is, in multisets of molecules. Instead of reaction rules that rely on substructures, their reactions operate on subsets of multi-molecules and convert them to other subsets of multi-molecules. Therefor all reactions have to be known at the level of whole molecules beforehand.

Their approach does not only look for a path to construct one molecule or one set of molecules, but they are interested in all paths between all possible multisets of molecules.

To limit the combinatorial explosion, they use some restrictions: They perform bidirectional search and limit the search depth to a fixed but arbitrary number. They only look at paths between multi-molecules that consist of up to m molecules, where m again is a fixed but arbitrary number. Note, however,

that within those generated paths multimolecules consisting of more molecules can appear.

1.1.7 The search for Reaction mechanisms

Clarifying the mechanism of a chemical reaction is a difficult task that can involve many experiments like crystal structure, the trapping of intermediates, kinetic studies, spectroscopic methods and so on. To support and guide experiments, computational studies are widely used.

However, computational chemistry in this context usually means quantum mechanical calculations, while there are fewer papers that approach mechanistic questions with other means of cheminformatics. In this section I will review some of them.

CAMEO

The CAMEO program[29] has different modules for different mechanism types. These are the basic/nucleophilic module, several acidic/electrophilic modules, the pericyclic module, the oxidation-reduction module, the free radical module and the carbene module. Each of the modules allows for several types of elementary reactions.

The CAMEO program has a perception module that estimates the pK_a value, the electrophilicity and nucleophilicity, aromaticity, energies, radical stability and so on. These properties are used by the reaction modules to identify reactive sites and rank them by reactivity. Thus main and side products of reactions can be identified. For reaction modules where the mechanistic steps are well known, elementary reactions are used to predict the course of a reaction. This allows CAMEO to predict unprecedented reactions.

As CAMEO can estimate the reactivity and the stability of different intermediates, only reaction steps with a high priority are actually performed. Therefor the problem of exponentially growing trees is circumvented. The electrophilic module, for example, performs additions of electrophiles to π-bonds only if no strong nucleophilic quenchers are present. Generated carbocations are allowed to rearrange to form the most stable one and only this one will further react.

Reaction Explorer - No electron left behind

J. Chen and P. Baldi[30] developed a program that uses a knowledge base of elementary reactions and their priorities under different conditions to predict and validate chemical reactions.

They extended SMIRKS[31] to explicitly encode the curly arrow notation and coded 1500 balanced and mapped elementary reactions as SMIRKS . Then they defined 80 different reagents or reaction conditions. For each reagent they supplied a list of possible elementary reactions together with a priority representing the reaction rates of these steps. For a given starting material and a given reagent their program would apply the elementary reaction with the highest priority that would match the starting material. Then it would take the transformed molecule as new starting material and repeat the process until an unreactive molecule has been reached.

The greatest strength of this approach probably also is its greatest weakness: There is a lot of expert knowledge coded in the lists of possible elementary reactions for different reagents and especially in their priorities. This ensures correct results in almost any case. Furthermore, prioritizing the elementary reactions makes any use of an heuristic obsolete, as only one reaction has to be applied at each step of the mechanism. The problem, however, as they themselves note in their paper, is that their approach, as any approach that depends on a manually created knowledge base, can only cover a small part of all chemistry known (or yet to discover).

A machine learning approach

The Reaction Explorer described in the section above was then used to generate a training set for a machine learning approach.[32]

Their model used an approximation of the MO theory to assign filled and unfilled molecular orbitals to atoms. The first machine learning step then learned to filter unproductive MOs. The remaining filled and unfilled orbitals were then combined in all possible ways to generate elementary reactions. A second machine learning step was used to rank the predicted reactions.

In an iterative approach the products of the top ranked reactions could be used as substrates for the next elementary reaction step.

1.2 Representation of molecules in the computer

One of the most basic requirements of all the approaches mentioned above is a good representation of molecules in the computer. In this section I will give a short overview over the most common ways to store molecules in the computer.

1.2.1 Molecules as graphs

Before I can describe the actual ways of storing molecules in the computer, I will give a short introduction to the mathematical model that is behind most of those storing formats. This is the model of a graph, which is also what all organic chemists use when they draw structural formulas.

A graph G is an object that consists of two sets, the vertex set $\boldsymbol{V}$ and the edge set $\boldsymbol{E} \subseteq V^2$.

Molecules can be represented as undirected graphs with labeled vertices and edges. The label (or coloring) of the vertices is the atom symbol together with optional additional information like charge and radical symbols. The edge label is the bond order. This representation allows to introduce special labels for aromatic bonds to avoid different resonance structures that represent the same molecule.

For a more mathematical definition of molecular graphs see Kerber *et al.*[33]

This representation is isomorph to a representation as multigraphs with up to three parallel unlabeled edges.

1.2.2 Connection table

One way to store graphs in the computer is the use of connection tables. A connection table, as used in the MDL mol files, consists of two parts: The atom block, a list of atoms, and the bond block, a list of bonds. In the atom block each atom is, explicitly or implicitly, assigned with an index. Besides the atom symbol additional information like charge, radical or isotope information can be present. In the bond block each line references two atoms with their index and assigns a bond label (bond order) to the bond between them.

```
4  3
 -5.3182    0.7145    0.0000 H   0  0  0  0  0  0  0  0  0  1
 -5.7307    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  2
 -6.5557    0.0000    0.0000 O   0  0  0  0  0  0  0  0  0  3
 -5.3182   -0.7145    0.0000 H   0  0  0  0  0  0  0  0  0  4
2  1  1  0  0  0  0
3  2  2  0  0  0  0
4  2  1  0  0  0  0
```

Figure 1.2: A connection table of formaldehyde (4 atoms, 3 bonds)

1.2.3 SMILES

The SMILES format is probably the most widely used format to store molecules and to exchange molecular data between different applications.[34]

History of SMILES

SMILES (Simplified Molecular Input Line Entry System) is a line notation for chemical molecules.[35] SMILES was developed by David Weininger and Arthur Weininger in the 1980s. It is intended as a compromise between the human and machine aspects of chemical notation. SMILES does not cover the three-dimensional aspects of a molecule but only the information that can be written down as valence structure, i.e. the information contained in the molecular graph.

SMILES was further developed by Daylight Chemical Information Systems[31]. Since 2007 the open smiles project has been developing a free version of SMILES[34].

How to write the SMILES-representation of a molecule

In SMILES strings atoms are represented by the atom label followed by extra information, together enclosed in square brackets. Hydrogen atoms can be given explicitly, omitted or put inside the square brackets of the atom they are attached to as in "[CH4]". The most common organic atoms like C, N, O and S can be written without square brackets, if no extra information is provided.

Single bonds are usually not written down explicitly, but consecutive atom symbols are interpreted as bonded by a single bond. "CCO" thus is the representation of ethanol. Double bonds are indicated by an equals sign (=) and triple bonds by a number sign (#). Side chains are enclosed in parenthesis. "C=C(O)C" thus stands for propen-2-ol. Rings can be broken at any bond to form a non-cyclic molecule that can be written as SMILES with the atoms next to the broken bonds marked by the same digit or a two digit number preceded by a percent sign (%). "C1CCCCC1" stands for cyclohexane. Lower case letters can be used to indicate aromatic atoms.

Example of SMILES strings

The following SMILES strings all represent morphine:

Oc1c(O2)c3c(cc1)CC(N(C)CC4)C5C43C2C(O)C=C5
CN1CCC23C4C=CC([OH])C2OC5=C3C(CC14)=CC=C5O
C123C(O4)C(O)C=CC1C(N(C)CC2)Cc5c3c4c(O)cc5
CN1[C@@]2Cc3ccc(O)c(O4)c3[C@]3(CC1)[C@@]4[C@@](O)C=C[C@@]32

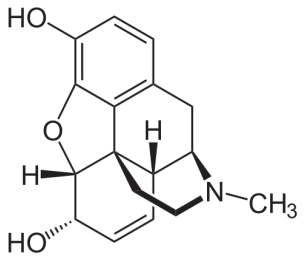


Figure 1.3: Structure of morphine

Canonical SMILES

As can be seen from the above example, several valid SMILES strings can represent the same molecule. This feature makes it a lot easier to write SMILES by hand without having to worry about the correct numbering of the atoms.

For some applications, however, a unique representation of a molecule is needed. This is accomplished by the use of canonical SMILES. Libraries like Open Babel can convert any given SMILES into a canonical form[36,37]. Also the GGL (*vide infra*) implements a writer for canonical SMILES. In order to generate a unique SMILES, the problem of deriving a unique numbering for a molecule has to be solved.[38]

1.2.4 Fingerprints

While the above mentioned SMILES and connection tables essentially store graphs, a different approach is used by fingerprints.

For many applications the exact structural formula of a molecule is not relevant. One is rather interested in the molecule's properties, like the presence of functional groups, the lipophilicity or the molecular mass. While the intuition of an (organic) chemist will hardly regard such a set of molecular descriptors as a representation of a molecule, it probably was all a chemist had 250 years ago.

Nowadays vectors of descriptors for molecules are used to allow for quick similarity searches against big databases, where (sub)graph isomorphism would be computationally too expensive. Fingerprints of descriptors can be precalculated and stored once when a molecule is added to the database. Whenever the user then submits a query molecule, these fingerprints can be used to find the most similar molecule from the database. For more information see the section on similarity (1.5.1).

1.3 Cheminformatics of reactions

Chemistry is the science of reactions between substances. Thus a representation of chemical reactions in the computer is vital in cheminformatics. Furthermore many applications require rules that allow the generation of new reactions for predictive chemistry.

In this thesis the term *reaction* will always refer to a chemical reaction that consists of educt molecules, product molecules and potentially a reaction mechanism. An example of a reaction would be "glutamic acid plus ammonia react to glutamate and ammonium".

The term *reaction rule*, on the other hand, refers to rules that decide whether a given set of molecules can react in a certain way and that can be used to generate the products, given the substrate. An example of a reaction rule would be "a carboxylic acid can react with an uncharged amine to generate carboxylate ion and a positively charged amine".

1.3.1 The chemical distance

The chemical distance (CD) between two molecules is the number of electrons that have to be shifted in a reaction to convert one molecule into the other. This corresponds to the number of bonds that have to be broken, changed with respect to their bond order, or newly formed. Often reactions proceed in the way that corresponds to the minimal chemical distance[22]. However, especially for multistep reactions that consist of several elementary reactions, the preferred course of the reaction may be completely different.

1.3.2 Representations of reactions

SMILES notation of reactions

Reactions can be easily written in SMILES notation by the use of the greater than sign (>)[31]. A complete reaction SMILES has the following general form: "Reactant>Agent>Product". The agent can be left out, be a solvent, a catalyst or perform any other function.

RXN files

RXN files store chemical reactions by storing two ore more connection tables for educt molecules and product molecules. These connection tables follow the specifications for mol files and are often referred to as mol blocks because they start with "$MOL". The line before the first connection table gives the information of how many mol blocks correspond to the educts and how many correspond to the products. Different mol blocks can be used for different educts, but it is also possible to put several connected components, i.e. molecules, into one mol block.

Within the atom block of the connection tables there is one column reserved for the atom map (*vide infra*). The atom map specifies which atom from the substrate molecules matches which atom from the product molecules. Atoms that have the same map index are mapped to each other.

The Imaginary Transition Structure

Fujita developed the *Imaginary Transition Structure* representation of chemical reactions[39]. This representation is generated by superimposition of the products' molecular graphs on the educts' molecular graphs to create one single graph for the reaction. In this representation there are three colors of bonds: *out-bonds*, *in-bonds* and *par-bonds*. These types of bond correspond to left-side, right-side and context bonds in the notation of GML rules (*vide infra*).

1.3.3 Classification of reactions

The reaction center

Based on his definition of the Imaginary Transition Structure, Fujita derives the *reaction center graph (RCG)*. The reaction center graph is derived from the Imaginary Transition Structure by deleting all atoms that are not attached to an in-bond or an out-bond and all bonds towards and between these atoms. This reaction center graph is a representation of what is sometimes called *reaction core*[16]. The *reaction graph (RG)* is derived from the reaction center graph by using unlabeled vertices instead of atoms[40]. *The basic reaction graph (BRG)* is created from the reaction graph by deleting all par-bonds from it.

These three levels of a graph representation can be used to classify chemical reactions that correspond to the same RG, RCG or BRG into the same class.[40]

The extended reaction core

The next more specific level in the hierarchic classification of Fujita would be the reaction center graph extended by all atoms that are one bond away together with the bonds connecting them to the reaction center.[40]

For their program Route Designer, Law *et al.*[16] propose a functional group completion algorithm to create an extended reaction core which should contain the reaction core and all functional groups that could probably influence the reaction. In contrast to an extension by shells of one ore two atoms, this approach is designed to only contain the relevant neighborhood of the reaction core.

A hierarchic classification according to the DU model[17]

The chemical distance (CD) covered by a chemical reaction can be used for the top level of a hierarchic classification. Reactions that cover the same chemical distance belong to the same CD class.

The next level is the classification according to the irreducible r-matrix in the DU model. To generate the irreducible r-matrix, all rows and columns that only contain zeros are deleted from a regular r-matrix. In this level of the hierarchy only the matrices are compared while the atoms that correspond to the rows and columns are not taken into account.

In the next level the atom symbols still are not taken into account, but the bonds and bond orders between atoms of the reaction core are used (rb-subclass). This corresponds to reaction graph. If the atom symbols are also matched, the next level (ra-subclass) is reached, which corresponds to a classification according to the reaction center graph.

The lowest level of the hierarchy finally is the level of individual reactions.

This classification can be used to asses the novelty of a reaction by searching for the highest level of the hierarchy at which no precedents for the reaction in question exist.

1.3.4 Reaction mapping

Balanced reactions

A reaction is called balanced if no atoms appear or disappear during the reaction. That means the sum of atoms in the products is the same as the sum of atoms in the educts. Furthermore the sum of atoms with each label must stay constant during the reaction except for special cases where a change in charge or adjacent hydrogens is specified as a label.

The Reaction mapping is not trivial

A reaction map is used to identify each atom from the educts of a balanced reaction with exactly one atom from the product side and vice versa. From the reaction map, clues about the mechanism of a reaction can be drawn. The reaction mapping is not always clear from educts and products and thus has to be specified explicitly. Consider, for example, the Diels-Alder reaction of unsubstituted diene with ethene to form cyclohexene. If you haven't heard about pericyclic reactions, several mappings could seam plausible:

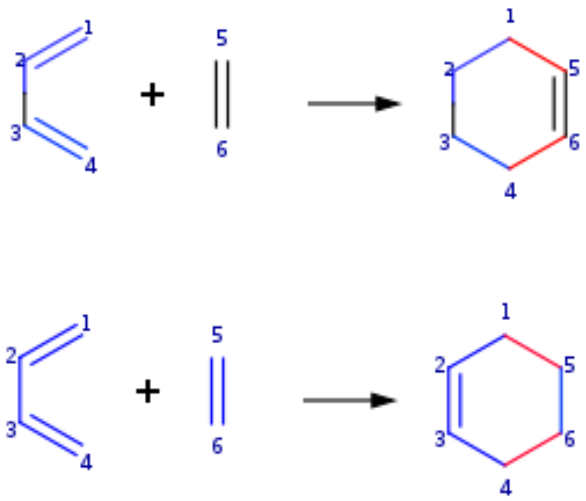


Figure 1.4: Different possible reaction maps for the Diels-Alder reaction. Bonds in red are newly formed, bonds in blue are changed in bond order. The second image is chemically correct.

Is the cyclohexene double bond the one that was in the ethene or is it one of the two double bonds from the diene? Both mappings imply a change of two double bonds to single bonds, the formation of two new single bonds and the breakage and reformation of at least one carbon-hydrogen bond.

The correct electrocyclic mechanism, on the other hand, corresponds to a map of the two carbon atoms in the middle of the diene to the sp^2 atoms in cyclohexene. This mechanism implies a change of three double bonds to single bonds, the formation of two new single bonds and the change of one single bond into a double bond.

Since no certain criteria exist to guess the correct map, it is often important to store the reaction map in chemical reaction data.

Uses of the atom map

The atom map is a necessary information for many applications. Arita[41] used it to infer pathways in the metabolic network of *E. coli* and showed that the small world property that used to be assumed did not hold when the flux of individual atoms was examined.

Furthermore the atom map strongly corresponds to the underlying reaction mechanism. If the reaction mechanism is known, many clues about the optimal conditions for the reaction can be drawn. Furthermore the classification of enzymes by their EC number is based on the mechanism of the reaction they catalyze.[42,43]

Finally the reaction map is necessary to generate reaction rules from the reactions. Beside synthesis planning algorithms, these reaction rules can also be used for artificial Chemistry.[44,45]

Reaction map in chemical file formats

The RXN files have a column in the atom block reserved for the atom map[46]. Atoms from different mol blocks with the same positive integer number for the atom map are mapped onto each other. In a SMILES string the atom maps are non-negative integer numbers which follow the atom label after a colon.

1.3.5 Algorithms to derive the reaction map

Several algorithms have been published that calculate the atom map which corresponds to the reaction path of minimal chemical distance. The chemical distance is usually defined as the number of bond changes weighted by their bond order. However, in many cases it is sufficient to only minimize the number of bonds formed or broken, while changes in bond order can be ignored[47]

The algorithms used to derive the atom map usually correspond to some sort of tree-search, as the optimal map has to be found among all possible maps.

Graph theoretical approaches to the atom mapping problem include finding the Maximum Common Edge Subgraph[48,49], subgraph matching of decomposition products on the substrate[50] or finding the largest set of largest subgraphs by the use of chemical cuts.[51–53]

Körner and Apostolakis proposed an approach that does not minimize the chemical distance but the energy of the imaginary transition state.[54] In contrast to the real transition state energy this imaginary transition state energy, is derived

by addition of the contributions of individual bonds. Thus their approach yields the solution corresponding to the principle of minimal chemical distance, if all bond weights are equal.

Mixed integer linear programming

First *et al.*[47] formulate an integer linear programming approach to this problem.

A linear program[55] consists of a linear objective function and linear equality or inequality constraints with many variables. The optimal value for all variables is sought for which the objective function assumes the maximal (or minimal) value and all constraints are fulfilled. The constraints define a feasible region that is a convex polyhedron in a space with as many dimensions as there are variables. The optimal value of the objective function in this polyhedron is searched for.

Linear programming is usually used for high-dimensional problems. There are optimizer programs that can solve arbitrary linear programs.

Mixed integer linear programming is a version of linear programming where one or more variables are restricted to integer (or even Boolean) values. These problems are more difficult to solve than normal linear programs because they are NP hard. Algorithms used for (mixed) integer linear programming problems are, among others, branch and bound algorithms.

In the approach of First *et al.*[47] there are four types of variables. The first type, which is used for the target function, accounts for the mapping of bonds from product to educt. The second type of variables stands for the actual atom map. The other two types are for the stereochemical part.

The target function reaches the optimal value if most bonds are mapped, while the constraints ensure that only atoms with the same symbol are matched.

Flamm *et al.* proposed a different linear programming approach that is based on cyclic transition states.[56]

Failures of the described algorithms

There are cases where the correct mapping cannot be derived by any of the algorithms described above. If the real reaction does not take the path of the minimal chemical distance (i.e. the minimal changes of bonds), no algorithm that optimizes the chemical distance will work.

In the case of enzyme reactions, structural reasons might pose a further complication. Take, for example, a proton transfer between two amino acid sidechains of a protein. Without structural data there is no way to determine whether the proton is transferred directly or whether a water molecule is deprotonated by one amino acid and reprotonated by the other in one step.

1.4 From reactions to reaction rules

Artificial and algebraic chemistry need rules that specify which molecules can react with each other and what products they will form.

The first generation of synthesis planning programs used reaction rules that were hand coded in special languages. This lead to the disadvantage that only a small part of all known chemistry was implemented in the programs.[16] Coding rules in an imperative language lead to a representation of rules that focused on the dynamic aspect of chemical reactions. Such rules were rather different to the individual reactions generated by a rule.

The more formal approach of Dugundji-Ugi uses a basis of reaction matrices representing electron changes that can be combined to express all known chemistry. The formal approach and the possibility to encode possible but not yet known reactions in their model is very promising. However, their model allows reactions with legal electron changes that still don't work in nature.

Due to the fact that nowadays an immense amount of chemical knowledge is stored in databases, different approaches to extract reaction rules from databases have been proposed.[15,16,26]

Route Designer[16] extracts rules by looking at the educts' and products' molecular graphs and the mapping between them. This approach leads to rules that are subgraphs of the educt and product side of the reactions they were extracted from. In this picture the rule is a generalization of the reaction. While the reaction is described by a graph of molecules and the changes that occur to them, the rule now consists of a subgraph and changes it should go through. Therefor this approach leads to two difficulties: If going from reaction to rule means going from graph to subgraph, one has to figure out how much of the original graph should be included into the rule's subgraph. In Route designer this is done by functional group completion.[16] Secondly a great challenge that still needs a lot of research is to find a way to generalize several similar rules (probably with the same reaction core - see 1.3.3) into a new, more general rule. This is of great importance, as there are hundreds, thousands or ten thousands of reactions stored in the individual chemical databases.

1.4.1 Chemical reactions as Graph Grammar Rules

Recently two systems have been developed independently that use graph grammar rules for chemical reactions: The Graph Grammar Library[57] and reaction MQL[58].

This work uses the GGL which in turn uses the Boost Graph Library[59]. The GGL has classes for subgraph matching and the application of graph grammar rules that can be used for all types of labeled graphs. Furthermore it has classes especially for molecular graphs to perform valence checks, aromaticity correction, energy estimation, etc.

1.4.2 The GML format for reaction rules

The graph modeling language GML[60] was extended by Martin Mann and Christoph Flamm[61] to allow for storage of graph transformations rules. In particular, chemical reactions can be seen as graph transformation rules.

The keywords "left" and "context" are used to specify the subgraph pattern that has to be present in the molecule to allow the reaction. This could be,

for example, a functional group. The "left" keyword specifies those features of the subgraph that change during the reaction, while the "context" specifies the features of the subgraph pattern that have to be present to allow the reaction, but that are not changed during the reaction. If no atomic properties change, all atoms are part of the context, since atoms are not changed during chemical reactions. Finally the "right" part of the rule specifies all features of the subgraph that will be created by the reaction. This means that all edges (bonds) in the "left" part will be broken, while all edges (bonds) in the right part will be formed by the reactions that follow this reaction rule.

Furthermore GML rules allow for wildcard labels to indicate that a certain vertex can match any atom label and constraints that disallow matching under certain circumstances.

1.5 Characterizing and scoring molecules

There are two ways of keeping the combinatorial explosion small in synthesis planning programs: One can apply a search strategy, where, depending on the strategy and the current synthetic target, only certain reactions are performed, while others are not performed or performed with a lower priority.[5] The alternative would be to use an heuristic that marks only some of the intermediate molecules for further exploration. Such heuristic probably needs to score a molecule by its similarity to a target, its complexity, its energy or some other property. In this section I will review some methods for scoring or characterizing molecules in the computer.

1.5.1 Similarity measures

Similarity measures between molecules were originally developed to allow for similarity search against big databases as an alternative to substructure search[62]. But when it comes to comparing molecules, there is no single, absolute definition of similarity, but it highly depends on the property of the molecule one wants to compare. For pharmaceutical research one may want to search for molecules with similar molecular mass, lipophilicity or other physicochemical properties. For most applications, however, a chemist is probably interested in some sort of structural similarity. Most approaches to similarity define a vector of descriptors for a molecule and a distance or similarity measure between these vectors.

Gasteiger *et al.*[63], however, used a different approach that is based on transforming molecules to more general representatives and assessing similarity between molecules, whenever identity between their more general representations was found. They proposed different measures based on different transformations like oxidizing all possible functional groups or extracting the carbon skeleton or the ring system from the molecules. They proposed a total of 21 possible similarity measures based on 21 different transformations. However, each of their similarity measures can only give two values for a pair of molecules, similar or dissimilar. Descriptor based similarity measures, on the other hand, can, depending on the descriptors, assign a continuous or almost continuous score (sometimes scaled to the interval between 0 and 1).

Several types of descriptors can be used for similarity measures: Counts of atoms or bonds, 2D fragments, 3D fragments, topological indices, counts of functional groups, etc.[62]. Depending on the similarity measure, the vector of descriptors can contain continuous numbers (with an optional scaling factor), discrete numbers or the Boolean values one and zero. The latter allows for a straightforward efficient implementation as bit strings. The Open Babel library[36] implements an approach outlined in the Daylight theory manual[31] that is based on fragments derived from the graph representation of the molecule. All linear subgraphs of length 1 to 7 are generated and hashed to a number, which is used to set a bit in the Boolean fingerprint vector.

Vectors of descriptors can not only be used for similarity calculation, but also to characterize reactions by differences in the descriptor vectors of products and educts (reaction vectors).[64]

Another alternative are similarity measures based on the maximum common edge subgraph[48] or similar approaches.

Similarity and distance coefficients

There are many coefficients available to calculate the distance or similarity between two (Boolean) vectors of descriptors[62]. Most common among them are the Minkowski distances (of which the Euclidean distance is a special case) and the Tanimoto coefficient. For Boolean vectors the latter is the number of bits that are present in both molecules divided by the number of bits that are present in at least one of the molecules. This gives a result between one and zero, where one means identity between the vectors of descriptors. The complement of the Tanimoto coefficient then is a measure for the distance between two vectors of descriptors and satisfies all conditions of a metric.[65,66]

Distances between multisets

While similarity and distance measures between single molecules are widely used in cheminformatics, little is published about distances between (multi-)sets of molecules despite the chemical distance. Unfortunately the chemical distance cannot be calculated without an atom mapping. As described in the section about atom mapping (1.3.4) , it is computationally expensive to calculate the optimal mapping. When fast calculation time is desired but the mapping is yet unknown, an alternative to the chemical distance has to be sought. Fortunately, mathematical models for distances between multisets of any kind of objects can usually be used.

Some distances, like the bag distance[67], are based on simply counting elements that are not present in both multisets. Others minimize some property over all possible multiset bijections or are based on the symmetric difference between two multisets.[68] Only some multiset metrics make use of the underlying distance between the multisets' elements.

1.5.2 Complexity measures

Bertz uses topological invariants on the molecular graphs to define indices of complexity.[69] The first of those indices, η, counts all different substructures consisting of two adjacent bonds that do not include any hydrogen atom.[70] Thus propane has a complexity of 1. Ethene has a complexity of one as well, because the double bond consists of two bonds and the path from the first C to the second and back again counts towards the complexity. Ethanone has a complexity of 3: The double bond from the oxygen to the attached carbon counts as two bonds, each of which is attached to the carbon-carbon-bond. The third substructure is the one in the double bond as in ethene. Ethyne has a complexity of 3 as well.

This complexity measure can be easily calculated by addition of contributions of individual atoms[71] in the center of the substructure. Atoms with two non-hydrogen bonds contribute 1 to the complexity, atoms with three bonds contribute 3 and atoms with four bonds contribute 6. As this method counts the contributions of double and triple bonds twice, the count of double bonds and 3 times the count of triple bonds has to be subtracted from the result.

For synthetic purposes extrinsic complexity measures[69] that take symmetry into account are needed. Furthermore the count of different hetero atoms contributes to complexity. Bertz defines C_η based on η to take symmetry into account, C_E for heteroatoms and C_T as a combination of both.

Other complexity measures defined by Bertz count the total number of different substructures of a molecule or the total number of substructures with only single bonds.[72]

Other complexity measures were proposed by Whitlock[73] and Barone[74], which are based on a more empirical approach, and by Boda[75], which relies on the different frequency of different substitution patterns in reaction databases.

1.5.3 Estimation of the energy

The Gibbs free energy of formation of chemical compounds is the energy that would be needed for the hypothetical reaction of its atoms in their reference elemental state to the molecule. As many elements are not too stable in their elemental state, the Gibbs free energy of formation is mostly negative.

It can be observed that the energy difference, which arises from the formation of a single bond, depends a lot on the atoms bound together and might as well depend on the neighboring atoms, but it does not depend on parts of the molecule that are very far away. Neglecting longer-ranged interactions and assuming the energy to be additive between the contributions of individual functional groups is thus a good approximation.

This idea lead to the development of group contribution methods that allow the estimation of the energy of molecules from their graph representation. Atomic coordinates or quantum-mechanical calculations are not needed for these (rough) estimations. However, a well defined decomposition scheme from arbitrary molecules to their fragments (for which an energy value is tabulated) is necessary.

Several research groups developed group contribution methods: Benson provided a hierarchy for additivity methods[76]: The molecular mass of a molecule is the sum of the atoms' contributions. For the energy of a molecule he provides an estimation based on the contributions of individual bonds and a more accurate method based on groups. In his group contribution model every atom has an energy contribution that depends on the directly bound neighbors. Mavrovouniotis[77] proposed a group contribution method that was specifically designed for aqueous systems.

Jankowski *et al.*[78] improved this method by calculating new values for each group together with an error estimate and by supporting more groups than were included in Mavrovounitis's work. He used multiple linear regression on a training set of 645 reactions and 224 compounds and calculated values for 74 molecular substructures. Furthermore he included 11 interaction factors to account for strain and electron delocalization. His approach is re-implemented in the GGL and was used in this work.

Chapter 2

Enzymes

2.1 Important milestones in the research of enzyme catalysis[79]

While there is still a lot of research necessary before we get a complete deep understanding of the catalytic power of enzymes in general, great progress towards this goal was made in the last decades.

Emil Fischer postulated the "key-lock-principle" (*vide infra*) in 1894 and was rewarded with the Nobel Prize in 1902.

The first experiments to systematically study enzyme catalysis were steady state kinetic studies that lead to the formulation of the Michaelis Menten kinetic in the 1920s. Two properties were used to characterize enzymes: the Michaelis Menten constant and the turnover number.

In 1926 the first enzyme was crystallized by James B. Sumner, who was rewarded with the Nobel Prize in 1946.

In the 1950s and 1960s the enhanced local concentration of substrate and catalytic residues in the active site was considered to be the main reason for the enormous catalytic power of enzymes. It was then shown that the restriction of rotation of substrate and catalytic residues played an important role. This was also the time when the induced-fit principle was proposed.[80]

The study of chymotrypsin lead to the first identification of a chemical intermediate derived from the substrate.

2.2 Key aspects of enzyme catalysis

2.2.1 Enzymes recognize the transition state

The key-lock principle and the induced-fit principle[81] describe a fundamental property of enzymes: With their active pocket and the side chains therein, enzymes recognize only certain molecules which are their substrates. The (older) key-lock principle says that the active site is complementary to the substrate and thus allows for binding only specific substrate molecules. The (newer) induced-fit principle recognizes that the active site without a bound substrate molecule does not have to be completely complementary to the substrate. However, once the substrate approaches the active site and finally binds, the enzyme changes its conformation in order to fit the substrate.

The conformational selection model finally accounts for the fact that enzymes, like all (macro) molecules, do not adopt only a single conformation, but are in an equilibrium between different conformations. In this picture, binding of the substrate simply shifts the equilibrium from one conformation of the enzyme to an other.[82]

However, as Pauling noticed already in 1948, in order to improve catalysis, the enzyme does not perfectly fit the substrate, but rather fits the transition state of the reaction it catalyses. When bound to the enzyme the conformation of the substrate is stressed and probably strained in a way that resembles the transition state. Thus the substrate's energy is increased, while the transition state's energy is reduced. This way the energy barrier of the reaction is reduced. In figure 2.1 the intermediate I_1 would be the enzyme-substrate complex, which has a higher energy than the sole educts. The overall transition state TS in the uncatalyzed reaction would then correspond to the transition state TS_2 of the enzyme catalyzed reaction, which is significantly lower in energy. Thus the activation energy EA is reduced, as seen in the figure. Note, however, that real enzyme catalyzed reactions can proceed via several transition states and intermediates.

However, the enthalpy contribution is probably the smaller factor compared to the entropy contribution. By binding the substrate tightly and fixing it in a conformation close to the transition state, the entropy of the substrate is reduced a lot. This loss of entropy means that most entropy is already lost when the substrate binds to the enzyme. Thus it does not have to loose much more entropy when it goes to the transition state. Since the transition state lies on a rather specific reaction path between two molecules, it always requires a rather specific conformation and thus a lower entropy.[82]

This loss of entropy when the substrate binds to the enzyme and the stress applied to the substrate is partly compensated by the binding energy of the substrate, but the energy of the substrate-enzyme complex is usually higher than the energy of dissociated enzyme and substrate. Thus, in equilibrium condition with an excess of substrate, less than half of all enzymes would actually bind to the substrate. However, this still facilitates catalysis, as the energy difference between the enzyme-substrate complex and the transition state-enzyme complex is reduced. Too tight binding between enzyme and substrate, on the other hand, would lead to a higher energy barrier.[82]

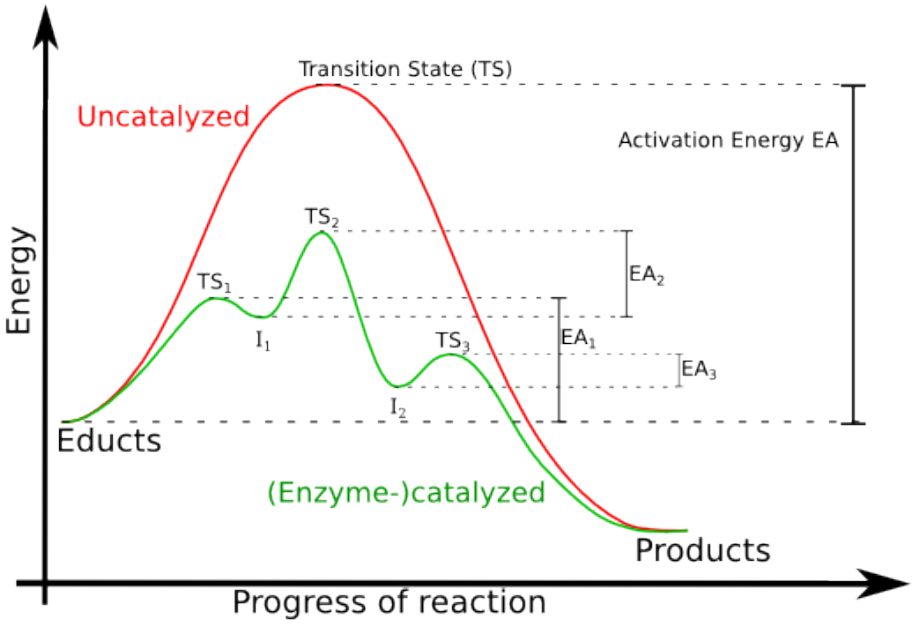


Figure 2.1: Energy diagram with and without an enzyme as catalyst.

2.2.2 Water is excluded from the active site

Another important aspect for many reactions is the fact that water can be efficiently excluded from the active site. Thus reactive intermediates that would immediately react with water can be stabilized. Therefor reactions can happen in enzymes that could not occur in water.

An example where protection of the active site plays an important role is the glutathione disulfide reductase, which can be found in MACiE (see next section) with the id number M0006. The active center is open to the solvent at two sides, one of which is the binding site of NADPH, the other of which is the binding site of the substrate. These two channels are separated by Flavin and a disulfide bond between two cysteine residues. Once NADPH and glutathione (the substrate) are bound in their respective pockets, they replace the solvent molecules. Thus the electron transfer from NADPH via the Flavin to break the Cys-Cys-disulfide bond is optimally protected from water. In the final steps of the mechanism glutathione is reduced while the Cys-Cys-disulfide bridge and the FAD cofactor are restored.[83] See figure 5.4 for a graphical representation of this mechanism.

2.3 The MACiE database

The MACiE[84] (Mechanism, Annotation and Classification in Enzymes) database collects information related to the catalytic mechanism of enzymes. The current release of MACiE[85], version 3, contains 335 distinct enzyme reaction mechanisms. They cover 321 distinct EC numbers and over 90% (182) of all EC sub-subclasses with a crystal structure available at the time of the release of MACiE version 3.[85]

Beside the elementary reactions that comprise the reaction mechanism, MACiE also contains information about the enzyme crystal structure and the catalytic domain. Furthermore annotation of cofactors, annotation of catalytic amino acid residues (both reacting amino acids and amino acids that electrostatically stabilize the transition state, annotated as spectator) with their function and a classification of the mechanistic steps are involved.

The database can be searched by the MACiE id number, the EC number, the catalytic domain CATH code and the PDB code. Furthermore similarity searches of protein sequences against MACiE are supported. For each elementary reaction similar entries can be found.

Of course MACiE contains links to the original research where the mechanism of an enzyme is published and to entries of the same enzyme in other databases.

2.4 Orphan enzymes

Enzymes for which a unique enzyme activity, but no genetic sequence is known are called "orphan enzymes." While the lack of genetic sequence information does not necessarily mean that the reaction mechanism is unknown, it still means that these enzymes are probably not very well studied.

The ORENZA[86] database contains a list of assigned EC numbers for which no protein- or genetic sequence is available in the TrEMBL and the Swiss-Prot database. Some of the enzymes found in ORENZA had their EC number assigned several decades ago. In that case it might be possible that the corresponding enzymes were rediscovered and assigned a different EC number but not merged with these old entries. There are, on the other hand, also some very new entries for enzymes where currently not much is known.

Part II

Methods

Chapter 3

From MACiE to reaction rules

Fur the purpose of this thesis, I wrote the program MechSearch that tries to figure out the reaction mechanism specifically for enzymatic reactions. As an input this programs expects a knowledge base of elementary reaction rules that can occur in enzymes. These reaction rules have to be given in the GML format and should contain only the chemically relevant environment around the reaction center.

In order to provide this knowledge base, I used the MACiE[85] database, of which Gemma L. Holiday generously provided me with a machine readable copy. A number of steps described in this section were necessary to go from the RXN files that came with MACiE to GML files[61] that fulfilled my needs.

3.1 Reading RXN files

RXN files[46] are rather easy to parse. Thus I wrote the Perl module readRXN.pm to read RXN files myself. This module currently only supports the more common version 2 RXN-files, while the newer version 3 is not supported. From the connection table this module reads the atoms and bonds with their labels, the charge information and, if present, the atom mapping. Any other information, including 3D coordinates, is ignored. Furthermore some properties are supported, while others are ignored. M CHG and M RAD lines are supported. As demanded by the file format specification, if a "M CHG" or "M RAD" line is present for charge or radical information, all charge information from the atom block is discarded.

While S-group lines are ignored, G and A lines can be read, if the flag $grouplabels is set to true. However, group abbreviation by G lines is not understood, instead G lines are treated as synonymous to A lines. If $grouplabels is true, they both are used to overwrite the atomic label with the group label or atom alias respectively. Note that within the MACiE RXN files, group labels are usually used to define a single atom to represent a group and not to abbreviate several atoms from the RXN file as a group.

3.2 The linear program used to determine the reaction map

Unfortunately, the atom mapping was missing or incomplete in most RXN files. Thus a version of First's[47] integer linear programming approach was used to calculate the reaction map.

3.2.1 From RXN to lp

I wrote a perl script to automatically generate the linear program from the RXN files. Since stereochemical information can not yet be represented by the GGL, we did not use the whole objective function as described by First *et al.*, but only the first two parts. As we had to deal with many aromatic systems, we chose to use the objective function that does not take bond order into account. However, to distinguish between the charged and the uncharged oxygen within a carboxylate-group, we added a third term that penalizes changes of charge with a weight half the weight of a bond change. We used the necessary constraints from First's model, but did not yet implement any tightening or symmetry breaking constraint. If a partial matching was present in the RXN file, it was used as an additional constraint.

To reduce the risk of wrong solutions and to speed up the solution process, we used the group label and atom alias information to overwrite the atomic label (that is, we set $grouplabels true when calling readRXN.pm). As atoms are constrained to match atoms with the same label, overwriting the atom label with the group label can determine the mapping of it right away, if the group label is unique. Indeed, MACiE uses unique labels for amino acid residues, as they use the three letter code together with the number of the residue as a label.

The modified version of First's[47] linear program we finally used was thus the following:

$$\begin{aligned}\zeta = &\sum_{(i,j)\in B^P}(1-\sum_{(k,l)\in B^E}\alpha_{ijkl})+\\ &\sum_{(k,l)\in B^E}(1-\sum_{(i,j)\in B^P}\alpha_{ijkl})+\\ &\frac{1}{2}\sum_{i,k\in A|C_i^E\neq C_k^P} y_{ik} \rightarrow \textbf{minimize}\end{aligned} \tag{3.1}$$

subject to (3.2)

$$\sum_{k\in A} y_{ik} = 1 \qquad \forall i \in A \tag{3.3}$$

$$\sum_{i\in A} y_{ik} = 1 \qquad \forall k \in A \tag{3.4}$$

$$y_{ik} = 0 \qquad \forall i,k \in A : T_i^E \neq T_k^P \tag{3.5}$$

$$\alpha_{ijkl} \leq y_{ik} + yil \qquad \forall (i,j) \in B^E \quad \forall (k,l) \in B^P \tag{3.6}$$

$$\alpha_{ijkl} \leq y_{jk} + yjl \qquad \forall (i,j) \in B^E \quad \forall (k,l) \in B^P \tag{3.7}$$

The variables are defined as follows:

$$A = \{1, 2, ...n\} \qquad \textit{Atom indices} \tag{3.8}$$

$$B^E = (i,j) : i,j \in A, i < j, i \text{ and } j \text{ bonded in educts} \qquad \textit{Educt bonds} \tag{3.9}$$

$$B^P = (k,l) : k,l \in A, k < l, k \text{ and } l \text{ bonded in products} \qquad \textit{Product bonds} \tag{3.10}$$

$$C_i^E \in \mathbb{Z} \qquad \forall i \in A \quad \text{charge of atom i in the educts} \qquad \textit{Educt charges} \tag{3.11}$$

$$C_k^P \in \mathbb{Z} \qquad \forall k \in A \quad \text{charge of atom k in the products} \qquad \textit{Product charges} \tag{3.12}$$

$$T_i^E \qquad \forall i \in A \quad \text{the type of atom i in the educts} \qquad \textit{E. atom symbols} \tag{3.13}$$

$$T_k^P \qquad \forall k \in A \quad \text{the type of atom k in the products} \qquad \textit{P. atom symbols} \tag{3.14}$$

Note that the letter C is used differently in First's paper.

The following variables are used in the linear program for the objective function and the constraints.

$$y_{ik} \in \{0,1\} \qquad \forall i,k \in A$$

$$1 \text{ if atoms } i \text{ matches atom } k, \text{ else } 0 \tag{3.15}$$

$$\alpha_{ijkl} \in [0,1] \qquad \forall i,j,k,l \in A$$

$$1 \text{ if bond } (i,j) \text{ matches the bond } (k,l), \text{ else } 0 \tag{3.16}$$

The variables α_{ijkl} can be declared continuous as they will only take the values 0 or 1 in the optimal solution due to the objective function and the constraints.[47]

The Perl script writes the linear program in the CPLEX .lp format.

3.2.2 Solving the lp

CPLEX[87] was used to solve the linear program for each of the RXN files. Initially a timeout of 10 minutes was given and after that the the currently best solution was written. If the time limit was exceeded (about 10 percent of all cases), the calculations were repeated with a longer time limit or the mapping was manually corrected. Those reactions files that were not balanced were detected and discarded at this stage, as the integer linear program corresponding to them was infeasible.

3.2.3 Writing the mapping into the RXN file

The mapping was then written back into the RXN files with a Perl script that uses regular expressions to extract the necessary information from the cplex solution file. As regular expressions are used instead of parsing the whole XML solution file, the script does not produce any error if the file is corrupted. However, errors would have been detected at a later stage of the file processing.

Alternatively the script also supports lpsolve[88] solution files. Thus if cplex is unavailable, lpsolve could be used instead.

Numbering schemes for the map index

According to the file specification the atom block of mol and RXN files has a separate column for the atom map. If two atoms from educt and product side of the reaction respectively map to each other, they have the same map index. If the mapping is not known for a certain atom, the map index is zero. The specification only says that atoms which map to each other should get the same number as map index, but it does not say which number this should be. Note that within the molblock of the RXN file atoms are referenced by their line number, not by the atom to atom map.

Since reactions can contain several molblocks at each side of the reaction it would be impractical to demand a specified numbering for the atom map. Thus I chose to use a numbering that seemed practical for my purpose. The atom map number in the RXN file can have 3 digits. I used the first digit of the mapping number for the number of the molblock at the educt side minus 1 (i.e. the first molblock has a zero as first digit) and the remaining two digits for the atom line within this molblock. If there were more than 99 atoms in a molblock, I used all three digits for the atom line, thus allowing only one molblock with up to 999 atoms. Since MACiE RXN files usually only contain one molblock per side of the reaction with several connected components in one molblock, both numbering alternatives yield the same result.

I chose this numbering that corresponds to the line in the atom block on the educt side because this made it easier to manually read the files and relate the bonds, charges and group lines to the corresponding atoms at least at the educt side.

3.3 From RXN to GML

The Perl script for conversion of the RXN files containing the atom map to GML rule files again used the readRXN.pm module, now of course with $grouplabels set to false. Thus all group label and atom alias information was ignored. This is important as the GGL requires valid atomic symbols for chemical graph grammar rules. The only other symbol allowed is the asterisk, which is the wildcard character that can match any atom. Thus any atom symbol "R" for (organic-) rest that was present in the RXN files was now converted to an asterisk. Furthermore the charge information was incorporated into the atomic symbol as in SMILES labels, but without the enclosing brackets.

3.3.1 Radicals in SMILES

For radicals, the SMILES standard does not specify a separate label, as radical information can automatically be deduced from the proton count that has to be given explicitly for radical atoms. This approach was not suitable for our application, as we wanted to specify rules, not molecules. Wherever a valence is missing in the subgraph pattern of a rule, any atom or group could be bonded in the target molecule. This could be prevented by constraining the number of adjacent atoms with the "ConstrainAdjacency" keyword, but it would be very unintuitive to use this for radicals. Avoiding constraints also made it easier to reverse the rules (*vide infra*). Furthermore in the case of molecules, the GGL can automatically add remaining hydrogens to complete the valence of an atom. At least here a special flag for radicals that should not be proton filled would be needed.

Using lower case letters for radicals, as it is sometimes proposed, would not work either, as it would interfere too much with ordinary aromatic labels.

Luckily, there is a better workaround. The SMILES specification allows for atom class labels within a SMILES string. They are specified with a colon at the end of the (complex) atomic label, followed by the class number. This number can, for instance, be used for the atom map within a reaction SMILES.

We, however, use these labels to specify the radical information. We use 5-digit numbers for this purpose. The first two digits are "90", an arbitrary label to indicate that now radical information follows. This allows us to distinguish radical labels from ordinary atom class labels right away. The next digit is the spin multiplicity of the radical. The final two digits are used to store information about the charge of the atom: The forth digit is used for the sign of the charge: 0 is for uncharged, 1 for a positive charge and 2 for a negative charge. The fifth digit finally is the amount of the charge.

It is necessary to incorporate the real charge into the radical 5-digit label, because the charge that is specified in a SMILES-like matter before the colon now has to be modified. The Graph Grammar Library performs a number of checks to assure the correct valence of an atom and can fill the missing protons. Internally the GGL knows nothing about radicals. Thus the charge before the radical label has to be modified to the charge the atom would have if it had the same bonds but no radical. This way we ensure it passes all consistency checks.

When it comes to the subgraph matching part and rule application, however, the atom class is not ignored. A "C-:90200" for an uncharged radical carbon only

matches an other “C-:90200”. If proton filling is applied to this atom, only three hydrogens are added, as would be to an ordinary negatively charged carbon.

3.4 Aromaticity correction

Whenever there is more than one valid mesomeric structure for a molecule, one would need more than one reaction rule to describe one reaction. To avoid this we described the compounds as aromatic structures with unique aromatic bond labels.

However, aromaticity describes more than only mesomeric structures. It describes a chemical property that is very hard to predict. Therefor it would be desirable to separate the problems of unique SMILES generation and aromaticity perception. To generate canonical SMILES regardless of aromaticity, a simple enumeration of mesomeric structures is desirable, which could be done by constraint programming.[89] When it comes to reaction rules, however, we are not only interested in a unique representation of SMILES, but also in the chemical properties of a molecule, as we don't want an aromatic reaction to happen to a ring with double bonds. Therefor we followed the classical approach and used aromaticity correction to generate unique SMILES.

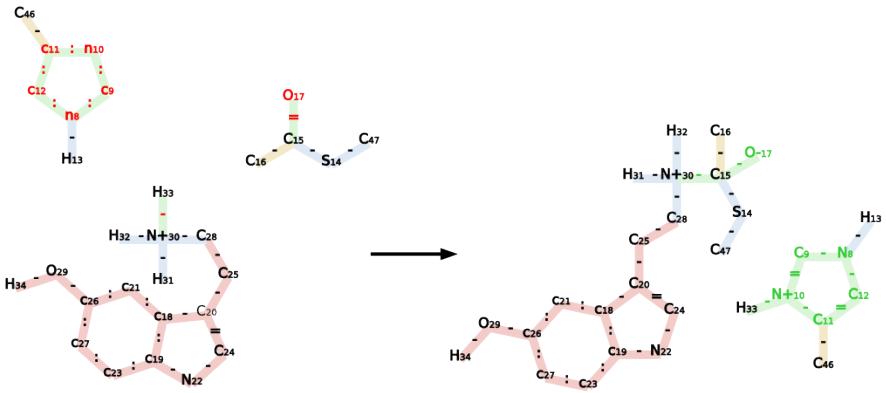
Since more than one model for aromaticity prediction exists, it is important to use the same model both for the rules one wants to apply and for the target graphs, i.e. molecules, on which the rules should be applied. The Graph Grammar Library GGL contains several aromaticity models, some of which are based on a machine learning approach on different learning sets. I chose the model "ChEBI:2011:pruned" that was trained on the ChEBI database.

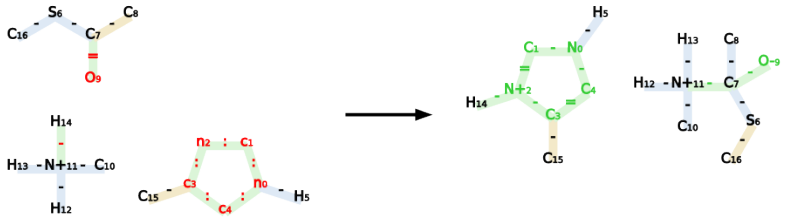
For aromaticity correction of rules I wrote the C++ program "ruleAromaticity". The name, however, is misleading, because strictly speaking you cannot correct the aromaticity of a rule. You can only correct the aromaticity of a molecule. Therefor my program converts the left side pattern and the right side pattern of the rule into two molecules. Then proton filling is performed for both molecules. This is necessary because MACiE contains only some explicit protons. Then aromaticity correction with the model "ChEBI:2011:pruned" is performed for both molecules independently. Finally a new rule is generated from the two molecules. The mapping between the atoms from the left and the right molecules was saved when they were generated from the rule and is now reused to generate the new rule. However, to make the rule less specific, all protons that were not present in the initial RXN files were not included into the new rule. Remaining valences of atoms in the rule's left side pattern allow for matching of structures with hydrogens or other atoms attached there.

Constraints from the initial rule could not simply be kept, since atoms or bonds might have changed their label. Reformulating the constraints would probably be possible, but was not implemented, as the rules generated from RXN files as described above did not contain any constraints anyway.

3.5 Finding the extended reaction core

The next step was to make the rules more general. To this end only the relevant part of the rules' context should be kept. Similar to the approach of Law et al.[16] I used the following algorithm: All neighbor atoms of atoms in the reaction core were examined. Heteroatoms, carbons bound to a heteroatom and carbons that were not sp^3 hybridized were included into the extended reaction core and their neighbors were examined in a recursive manner. The neighboring sp^3-carbons with no heteroatoms attached were not investigated. While these carbons themselves could theoretically be added to the reaction core with a special flag, this was not done for the rules used in this thesis.





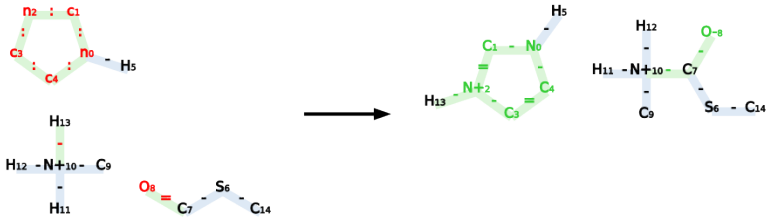


Figure 3.1: Extension of the reaction core. The reaction core is shown in light green. Parts shown in light blue are functional groups precieved by the algorithm. The yellow parts are part of the extended core in Ruleset50, as they have a distance of one to the extended core. Red parts of the context were not used in the rules. Parts of this image were automatically generated with chemrule2svg.pl, which is distributed together with the GGL[57].

3.6 Filtering of rules

Some rules had to be discarded because they contained an error or chemistry that was not compatible with the GGL. Table 3.1 lists all discarded rules together with the reason for discarding them.

MACiE ID	Stages	# rules	Reason for discarding
M0004	1-3	3	Complex
M0008	2,4	2	Text label "Base" involved in reaction
M0013	8-9	2	Text label "AmiOx"/"AmiRed" involved in reaction (electron transfer)
M0014	1-7	7	Complex
M0020	1	1	Text label "HEC" involved in reaction (electron transfer)
M0020	2	1	The proton filling of the GGL adds a wrong number of H to a complicated aromatic ring
M0033	1-3	3	Complex
M0034	1-7	7	Complex
M0037	1-3	3	Complex
M0037	4	1	Not balanced
M0038	5-11	7	Not part of public release of MACiE
M0043	1,3	2	Complex
M0052	4	1	Not balanced
M0062	1,6	2	Complex
M0063	1,4,6	3	Complex
M0068	2-3	2	Text label "ETFox"/"ETF1e" involved in reaction (electron transfer)
M0070	2	1	Text label "Base" involved in reaction
M0072	1	1	Not an elementary reaction, not enzyme catalyzed
M0077	4	1	Not balanced
M0080	2	1	Two elementary H-transfer reactions in one file
M0087	2	1	Complex
M0089	1-7	7	Carbocation
M0095	5	1	Not balanced
M0099	4-5	2	Not balanced
M0102	2-5	4	Text label "b2heme(ox)"/"b2heme(red)" involved in reaction (electron transfer)
M0105	1-2	2	complex
M0105	3-4	2	Text label "Acceptor"/"reducedAcceptor" involved in reaction (electron transfer)
M0106	6	1	Not balanced
M0107	1-7	7	complex
M0111	9-10	2	Text label "Ferrdoxin"/"Ferrodoxin-" involved in reaction (electron transfer)
M0114	2-3	2	Text label "ETFox"/"ETF1e" involved in reaction (electron transfer)
M0117	4-5	2	Text label "Hemoprotein"/"Hemoprotein-" and "Hemoprotein2-" involved in reaction (electron transfer)

MACiE ID	Stages	# rules	Reason for discarding
M0119	3-7	5	carbocation
M0121	1-4	4	complex
M0122	2	1	valence change is not yet supported by the GGL
M0123	1-2	2	Text label "Donor"/"Donor1-"/"oxidizedDonor" involved in reaction (electron transfer)
M0123	5	1	valence change is not yet supported by the GGL
M0124	1-11	11	Too many water molecules
M0124	1+03- 05	4	complex
M0125	1+03	2	complex + redox reaction with metal ion
M0125	4+05	2	complex + redox reaction with metal ion
M0126	1-5	5	complex + redox reaction with metal ion + Text label "acceptor" involved in reaction (electron transfer)
M0127	1-6	6	complex + redox reaction with metal ion
M0128	5	1	reaction not balanced - "hv" appearing as atom
M0129	1-6	6	complex + redox reaction with metal ion
M0130	2-9	8	complex + redox reaction with metal ion + Text label "Fe(III)2S2" involved in reaction (electron transfer)
M0132	1	1	Text label "Base" for general base
M0132	6	1	not balanced - "hv" appearing as atom
M0133	1-6	6	complex
M0134	1-5	5	complex
M0135	1-6	6	complex
M0136	1-5	5	complex
M0137	1-7	7	complex
M0138	1-2	2	complex
M0139	1-3	3	complex
M0141	4-5	2	Text label involved in reaction (electron transfer)
M0141	9-10	2	Text label involved in reaction (electron transfer)
M0142	2-3	2	Text label "Adrenodoxin" involved in reaction (electron transfer)
M0143	2	1	complex
M0144	1-5	5	complex
M0145	1-5	5	complex
M0146	1-2	2	complex
M0146	5-9	5	complex
M0153	1	1	valence change is not yet supported by the GGL
M0156	1-3	3	complex
M0156	5	1	complex
M0166	2-3	2	carbocation and oxoniumion
M0176	1-2	2	complex

MACiE ID	Stages	# rules	Reason for discarding
M0183	1-2	2	identity reaction (Excitation energy transfer only)
M0186	2+06	2	not balanced
M0189	1	1	identity reaction (isomerization of the peptide bond conformation)
M0190	1-2	2	carbocation
M0192	1-2	2	oxoniumion
M0200	2-3	2	carbocation
M0201	1	1	Text label "Base" for general base
M0208	3-4	2	Text label "Fe(II)Fe(III)S2" and "Fe(III)2S2" involved in reaction (electron transfer)
M0208	6	1	Text label involved in reaction (electron transfer), redox reaction with metal ion
M0212	1-15	15	complex (Fe-S-cluster)
M0216	1	1	complex
M0223	1	1	Text label "Base" for general base involved in reaction
M0225	4+11	2	not balanced
M0226	3	1	valence change is not yet supported by the GG
M0231	1-6	6	complex, redox reaction with metal ion
M0233	1-2	2	complex
M0237	2	1	not balanced
M0239	1-5	5	complex involved in reaction, Text label involved in reaction (electron transfer)
M0239	8	1	Text label "R+."/"R" involved in reaction (electron transfer)
M0240	4-5	2	carbocation
M0247	5 1	1	unbalanced
M0250	1-6	6	complex

Table 3.1: Rules that had to be discarded.

The main reason for discarding rules was chemistry that was not compatible with the current version of the GGL: Coordinative bonds cannot be represented by the GGL. If they are coded as normal single bonds, however, the molecules do not pass the valence check of the GGL. Starting with M0051 the valence check was not performed before, but after aromaticity correction and extraction of the extended reaction core. Thus complexes that are not part of the reaction core are no problem, while only reactions with a coordinative bond in the reaction core had to be discarded for higher MACiE id numbers. Another type of incompatible chemistry are carbocations. Right now the proton filling algorithm would add H atoms to a C^+ atom until it had a valence of five (instead of three as would be correct in most cases). This happens as a wrong analogy to N^+ and O^+, where the positive charge is linked to an additional valence.

Both problems will be taken care of in one of the future versions of the GGL[90].

In the case of M0020, step 2, the flavin part of FAD is perceived as aromatic by the GGL. For atoms with two aromatic bonds, the proton filling algorithm assumes a contribution of three to the valence by these two bonds. Unfortunately,

in the case of flavin, one aromatic nitrogen atom has two carboxy-C atoms as neighbors. Thus the aromatic bonds can only contribute a value of 2 to the valence. It would thus be correct to add two protons to this nitrogen, while the GGL only adds one.

Another reason for discarding rules were unbalanced reaction files and reactions that involved text labels. The latter mostly occurred for electron transfers where it is unpractical to draw the complete atom structure of a big electron acceptor/ donor. Thus only text labels like "ETF" or "HEC" were present in the MACiE files. As long as these text labels were only part of the rule's context, they were no problem as we extracted the extended reaction core of the reaction anyway. However, when electrons were transferred from or to these entities, this chemical transformation could not be coded as a GGL rule.

3.7 Clustering the rules

To avoid duplicates, a first clustering of the rules was performed. Whenever the chemical transformation of one rule could be achieved with another rule that was equal or more general, only the more general rule version was kept. For rulesets (*vide infra*) where all rules were treated as reversible, it was checked if the rule in either direction could be expressed by the more general rule.

The algorithm to do this was the following. For both rules a canonical SMILES representation of the reaction center imaginary transition state was generated. If these SMILES were equal, subgraph matching for the whole rule graph was performed. The more general rule had to be a subgraph of the more specific rule. For the purpose of subgraph matching the rule side information (i.e. left side, context or right side for edges, context or label change for nodes) was incorporated into the edge- and node-labels respectively. Again, constraints were ignored, as I did not work with rules containing constraints.

This algorithm makes use of the following idea:

Definition: A rule (consisting of only left, right and context part without constraints) is *more general* than a *more specific* rule if the rules are not identical and the more general rule can be applied to all substrates of the more specific rule and if it generates the same chemical transformation (including the atom map).

Definition: The *left side pattern* of a rule (as before without constraints) is the graph described by the left side and the context of the rule.

Lemma: A rule is *more general* than another rule if the nodes of one rule can be numbered in a way that left and right part of both rules are equal while the context of the more specific rule consists of the context of the more general rule plus additional edges and/or vertices.

Proof: The more general rule's left side pattern is a subgraph of the more specific rule's left side pattern. A rule only matches a (multi-)molecular graph if its left side pattern is a subgraph of this (multi-)molecular graph. The *is subgraph of* relation is obviously transitive. Thus the more general rule will always match on the same position where a more specific rule matches. As left and right side of both rules are equal, it generates the same graph transformation.

□

3.7.1 Limits of the current approach

Cases where the context of one rule is a subgraph of the other rule are relatively simple. The situation is more complicated in cases where the context of two rules consists of the maximal common edge subgraph and different attachments for each rule (see figure 3.2). The SYNCHEM group uses a generality and a specificity bound where the generality bound is the reaction core unless examples of reactions that cannot work are known to the program.[15] Simply omitting all parts of the context that are contradictory between several rules would make the rule too general, unless data was present to make the generality bound more specific.

In the framework of the GGL, constraints could probably be used to generalize some reactions. However, this was not done during this theses but remains as a challenge for the future.

Another problem can be incorrect atom mapping. If two identical reactions have a different atom mapping, they are correctly not clustered together. This leads, however, to more rules than necessary. If both atom maps correspond to the same objective value in the linear programming approach used to derive the atom map, it can happen that from two identical reactions two rules with different atom maps are generated. In that case the ruleset contains wrong redundancies.

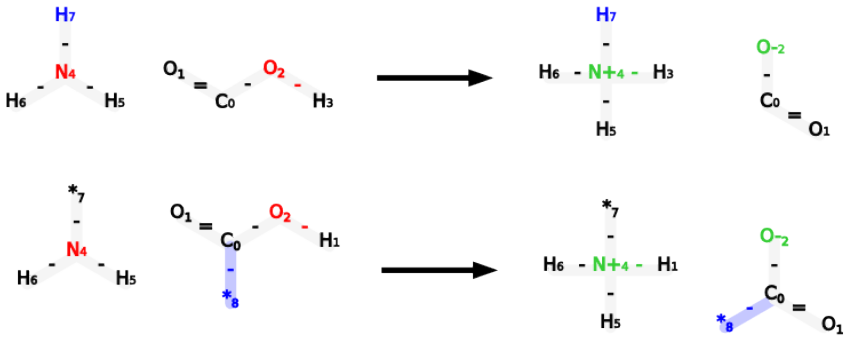


Figure 3.2: An example of two rules that sometimes generate the same reaction but cannot be ordered according to the *more specific* relation. The blue parts are positions where the one rule is more specific than the other. The examples are from MACiE id M0060.stg06 and M0176.stg03. The star character is a wildcard label. Parts of this image were automatically created with chemrule2svg.pl, which is distributed together with the GGL[57].

3.8 Reversal of rules

Most chemical reactions and especially most elementary reactions within enzyme active sites are reversible. Therefore a way to reverse a GML rule would be useful. Furthermore, to perform a bidirectional search reversal of rules is necessary.

To reverse a rule, the left and the right part of the GML file simply have to be switched, if no constraints are present. With constraints, however, reversal of rules is a difficult problem, because more rules might be necessary to represent the reversal of a single rule. This problem was not solved in this work, as I only had to deal with rules that contained no constraints.

3.8.1 The problem with aromaticity

As aromaticity correction is usually performed on the reaction products right after the rule application, there is a little complication: Consider the case of a protonated nitrogen within a ring. A rule could make this nitrogen lose its proton and change it to an aliphatic uncharged nitrogen. However, the ring could change its properties and aromaticity correction could change the nitrogen label to an aromatic label. Then the reverse reaction rule would not match, as aliphatic nitrogen does not match aromatic nitrogen.

Unfortunately this problem could not be overcome. As rule sides describe subgraph patterns it can always happen that an unexpected molecular fragment attached to the rule's subgraph pattern changes the aromaticity of the product in either direction. To avoid this problem, one would therefor have to include an aromatic and an aliphatic version for every rule. This would, however, lead to a loss of information, as in real chemistry it may depend on aromaticity whether or not a reaction can take place.

Aromaticity correction, on the other hand, can not be reversed. It would theoretically be possible to remove all aromaticity information and generate one of several possible resonance structures. This, however, would not really serve our purpose.

Therefor there might be cases where the results of the program depend on the search direction and bidirectional search is not 100% bidirectional. As there are only very few cases where this problem actually occurs, this should not be too big a problem.

The reversed rules were generated whenever needed by the program and were not saved to a file.

3.9 Sets of rules

From the rules that remained after filtering, several sets of reaction rules were designed. Clustering, as described above, was applied to all rules within the set.

3.9.1 Ruleset50

The first set of rules, "Ruleset50", contains all rules for MACiE reactions with a number smaller or equal M0050. Furthermore 6 hand coded rules were added for all combinations of proton transfers from ROH, RSH or R_3NH^+ to RO^-, RS^- or NR_3. These hand coded proton transfer rules were added because only some proton transfers that are chemically easily possible occur in the first 50 reactions of MACiE. For this ruleset, all reactions were considered to be reversible.

During extraction of the extended reaction core for rules of Ruleset50, functional group completion was performed. Furthermore all atoms with distance one to the reaction center (that is, atoms attached to the reaction center) were included regardless of their involvement in functional groups. That made the rules of Ruleset50 a little more specific than rules from other rulesets.

The special feature of Ruleset50 is the fact that the atom mapping was hand checked for all elementary reactions within this rule set.

The first clustering to remove duplicates yielded 135 unique rules that could represent the initially present 162 rules.

3.9.2 Ruleset100

"Ruleset100" contains all rules for MACiE reactions with a number smaller or equal M0100 plus the 6 hand coded rules used in Ruleset50. Again all rules were treated as reversible. During the extraction of the extended reaction core only functional group completion was performed, but atoms in distance one were not automatically added to the extended reaction core, as was done for Ruleset50. Ruleset100 consists of 295 unique rules representing a total of 346 rules.

3.9.3 Ruleset250

"Ruleset250" contains all rules for MACiE reactions with a number smaller or equal M0250 plus the 6 hand coded rules used in Ruleset50. Again all rules were treated as reversible. Ruleset 250 contains 711 unique rules that represent a total of 884 rules.

The rules used in Ruleset250 are even less specific than those from Ruleset100 as sp^3 carbon atoms attached to aromatic rings now were not treated as functional carbons during functional group completion. This difference between the three rulesets during reaction core extension was due to the following fact: At the beginning of my work for this theses, I was worried rules would become applicable in cases where chemical reactions would not really occur if too little atoms were included into the reaction core. After several calculations, however, it was seen that only a very small part of all chemistry could be generated with these more restrictive rules. Unfortunately it often depended on chemically unimportant atoms or groups whether a reaction could be applied or not. Thus for rulesets that were generated later during my work, functional group completion was modified in a way to make the rules less specific.

If larger databases of some 10,000 reaction steps were available, it would probably be good to run functional group completion in a way that makes rules more specific. This would be especially necessary due to the fact that no examples of failed reactions are present in such databases (*cf.* the generality bound approach of the SYNCHEM group[15]). For Ruleset250, on the other hand, where less than thousand reactions were available to extract the rules from, the approach that generates more general rules was definitely preferable.

Chapter 4

MechSearch

4.1 Assumptions and approximations

The program MechSearch uses a graph based model for substrates, potentially reactive residues and cofactors. It calculates possible reaction mechanisms as sequences of elementary reactions that lead from the substrate to the product. MechSearch is designed to require very few knowledge about the enzyme that is to be studied. Therefore some approximations and assumptions are made:

1) All absolute and relative positions of residues and substrate groups are ignored. This is a very rough approximation, as it is well known that in reality proximity between reactive groups plays an essential part in enzyme catalysis. However, we choose to ignore this type of information, since we want our program to be applicable even when the geometry of the active site is not known. On the other hand, the results of calculations with our program might make it possible to propose a plausible geometry for the active site. For more detail on this aspect see the discussion and outlook sections.

2) Only a defined number of copies of each substrate or residue can occupy the active site at the same time, as there is only limited space available. Typically this is one copy of the substrate and up to three copies of amino acid residues. While this assumption probably seems obvious, it is the main difference between enzymatic reactions and reactions in solution within our model. It also means that we are dealing with small numbers of molecules and not with concentrations. For example: If the substrate molecule is changed by a reaction, we do not allow the modified version of the substrate to react with another copy of the substrate (modified or unmodified).

3) After the overall reaction the enzyme is restored to its original state. All residues that are modified in the course of a reaction have to be changed back again. This is essentially the definition of a catalyst.

4) We model chemical reactions as graph grammar rules. We assume that chemical reactivity can be very well described by subgraph patterns. All geometric information about the atoms and groups within a molecule is thus ignored and steric hindrance cannot be accounted for.

5) The chemical space of enzymatic reactions is very big and diverse, but still is probably only a subspace of all possible chemical reactions. While it is up to the user of the program to choose the allowed chemical transformations, we only allowed transformations that had precedents in enzymes and were generated as described in the section before. We tested our program with different rule sets.

6) We do not model the pKa of bonds to hydrogens or the nucleophilicity or electrophilicity of reacting groups. Thus all reactions that are allowed from the reaction rules are applied without an estimation of the reactivity of the functional groups within the specific molecule. We assume that enzymes were able to modify the pKa and the reactivity in the desired way anyway.

Altogether especially the first approximation makes this a rather crude model that neglects several aspects of enzyme catalysis. However, as long as no heuristic is applied, this leads to mathematically well defined results that are based on simple principles. Furthermore, this allows to examine a special aspect of enzyme catalysis that sometimes is neglected: The chemical reaction pathway as a sequence of elementary reactions.

4.2 Definition of multisets

Definition: A *multiset* $\boldsymbol{A}$ is a generalization of a set in which identical elements can occur multiple times. One can thus write a multiset as a tuple $(\boldsymbol{S}, m)$ where $\boldsymbol{S}$ is a set of elements and the function $m : \boldsymbol{S} \to \mathbb{N}_0$ is the multiplicity of these elements within the multiset.

Definition: The *union* between two multisets $\boldsymbol{A}$ and $\boldsymbol{B}$ is a multiset for which the multiplicity of each element is the maximum of its multiplicities in $\boldsymbol{A}$ and $\boldsymbol{B}$. We then write $\boldsymbol{A} \cup \boldsymbol{B}$.

Definition: The *multiset sum* between two multisets $\boldsymbol{A}$ and $\boldsymbol{B}$ is a multiset for which the multiplicity of each element is the sum of its multiplicities in $\boldsymbol{A}$ and $\boldsymbol{B}$.

Definition: A multiset $\boldsymbol{S}$ is called a submultiset of $\boldsymbol{A}$ if for each element in $\boldsymbol{S}$ its multiplicity in $\boldsymbol{A}$ is at least as high as in $\boldsymbol{S}$.

Definition: The intersection between two multisets $\boldsymbol{A}$ and $\boldsymbol{B}$ is a multiset for which the multiplicity of each element is the minimum of its multiplicities in $\boldsymbol{A}$ and $\boldsymbol{B}$. We write $\boldsymbol{A} \cap \boldsymbol{B}$.

Definition: The multiset difference $\boldsymbol{A}$ *without* $\boldsymbol{B}$ is the multiset C for which the multiplicity of each element is the maximum of 0 and the multiplicity in $\boldsymbol{A}$ minus the multiplicity in $\boldsymbol{B}$. We write $\boldsymbol{A} \setminus \boldsymbol{B}$.

Definition: Two multisets are equal ($\boldsymbol{A} = \boldsymbol{B}$) if all elements that occur in any of the two multisets with a multiplicity of at least 1 have the same multiplicity in both multisets.

4.3 States instead of molecules

As mentioned above, only a certain number of instances of a molecule can coexist in the active site of an enzyme. This leads to the definition of states.

Definition: A *state* is a multiset of molecules that coexist at a given time in the active site of an enzyme. Similarly to *states* in our system, Félix *et al.*[28] use the term *multi-molecule*.

Definition: A state $\boldsymbol{S}$ is called a *substate* of $\boldsymbol{A}$ if and only if the multiset $\boldsymbol{S}$ is a submultiset of the multiset $\boldsymbol{A}$.

Definition: The *overall educt state* is the state of the system before the first reaction happens. It consists of the substrate(s), cofactors and amino acid residues.

Definition: The *overall product state* is the state of the system after the enzymatic reaction. Products are present instead of the substrate(s), cofactors may have been converted and the amino acid residues are restored to be the same as in the educt state.

In the implementation in MechSearch, the overall educt state and the overall product state are specified by the user as input.

All other states that will be generated during program execution are called *intermediate states*. These states are iteratively generated by bidirectional search from the overall educt state and the overall product state.

Definition: Two states are chemically *compatible* if atom mapping between them is theoretically possible, that is, if for each atomic number both states contain the same number of atoms and if the sum over all charges is the same in both states.

For each reaction mechanism, we only investigate a subset of all states compatible to the overall educt state. In particular, we demand that the product state has to be compatible to the overall educt state.

Note, however, that until now the conservation of charge between overall educt state and overall product state is not checked by the program MechSearch.

Definition: A *chemical reaction* is a transformation that takes a number of substrate molecules and converts them into a number of product molecules. Within our model of algebraic *Corollary:* If the overall educt state and the overall product state are compatible, all intermediate states will thus be compatible as well, as they are generated by (a series of) chemical reactions from the overall educt or overall product state.

Definition: A *reaction rule* is a generalization of chemical reactions. It consists of a substrate subgraph pattern, a product subgraph pattern and a mapping between them.

Definition: A *reaction path* is a sequence of $k+1$ states together with a sequence of k reactions, where the ith reaction can be used to convert the i^{th} state to the $i+1^{th}$ state. For our purpose we are usually only interested in reactions paths between the overall educt state and the overall product state, that is, reaction paths where the 1^{st} state is the overall educt state and the last state is the overall product state.

4.4 Implementation

MechSearch was implemented in C++11[91] and uses object orientated programming.

The `StateStorage` class relies on containers from the Standard Library[92]. Only one instance of this class is necessary and it is used to store the states as defined in the section above in an ordered way. Exploring one state after the other in a breadth first search corresponds to iteration through all states stored in the instance of the `StateStorage` class. This class is, however, designed to allow to easily change the order of states in a way that implements a priority queue. Associated with the states several state properties are stored, among which are the iteration depth and the reactions that generated this state together with the respective precursor states. When a new state is generated by a chemical reaction, the `StateStorage` class automatically checks if this state already exists. If it does, instead of adding a new state, the properties of the existing state are modifies to store an alternative path that generates it.

The `SmilesMap` class is also used in one instance only. It is used to store a canonical SMILES string and a graph object (as a `ggl::chem::Molecule`, which is a typedef to a `boost::Graph`) for each molecule that occurs during the search. Each molecule is associated with an index (although it would be possible to cluster several molecules - e.g. tautomers or mesomeric formulas if no aromaticity correction is performed - to one index). That index is used in the `StateStorage` class to store the individual states.

The `ReactionSaver` class is used to store all reactions that were performed as transformations of one substate to another. This way, when the same reaction rule has to be applied to the same substate (which now is a substate of a different state), subgraph matching does not need to be performed again, but all stored reactions can be applied by simply removing the reaction's educt substate from the state and adding the reaction's product substate to the state.

The rules and SMILES that were input by the user are parsed using the parser classes from the GGL. If the reaction is not balanced, the program tries to balance it by adding a minimal number of H_3O^+, H_3O^+ and OH^-. If a reaction can not be balanced this way, the program returns an error message and exits.

The `main` routine then iterates through all states until a given search depth and applies all rules to them. Rule application is done in a similar fashion as written by Martin Mann for toyChem[57]: If a left side pattern of a rule consists of several connected components, a match for each component is searched for first. Only when each component is matched by a molecule, all molecules are put together into an overall target graph and the whole subgraph pattern is matched on them. Subgraph matching and rule application is done with the GGL. The main iteration consists of the following steps:

1. The next state to explore is retrieved from the `StateStorage` class and becomes the new current state.

2. If a heuristic is used, it is now called and allowed to mark the current state or any other that is later in the search queue as uninteresting.

3. If the current state is flagged as uninteresting, execution is continued at step 1.

4. The program iterates through all reaction rules.

5. For each component of the rule's left side pattern, all possible matching molecules of the state are found. All substates where each molecule matches one

component of the rule are generated.

6a. If the substate is prestored for this reaction in the `ReactionSaver` class, the rule can be quickly applied.

6b. Otherwise the reaction is applied via subgraph matching.

7. The new states that were generated by the reaction are stored in the `StateStorage` class. This class usually puts them at the end of the search queue.

4.4.1 Sampling of paths

When the iteration is finished, the properties associated with the states are used to generate the whole reaction network as a `boost::Graph`. Then boost's implementation of Dijkstra's shortest path algorithm[93] is used to find the shortest reaction path. All paths up to a certain length given by the user as an optional argument can be enumerated by the program via a depth first search. This procedure is quick for very short maximal path lengths but is computationally too expensive for longer path lengths. Thus two methods to sample some interesting paths can be used:

The first sampling method used Dijkstra's shortest path algorithm to calculate the shortest path with all edge costs set to 1. Then the cost for all edges in the cheapest path is doubled and the new cheapest path is found. If this procedure is repeated several times, a sample of different interesting paths is generated. The second method assigns random costs to each edge and then used Dijkstra's shortest path. Again this procedure can be repeated several times to generate different paths. The disadvantage of both methods is that often the same path is found in different iterations. Thus the number of generated different paths is smaller than the number of iterations. While these methods may give a first impression of some interesting paths, they probably don't yield all interesting reaction paths. Thus additional programs were developed to analyze the output of MechSearch (*vide infra*).

4.5 Heuristic

In the search for a heuristic, several calculations were performed and the properties of different states were inspected (see results and discussion). Finally it was decided to use a heuristic that consists of two parts.

4.5.1 Heuristic part 1

The first part of the heuristic looks at the elements that are only found in either the educt state or the product state, but not in both.

Definition: In the following the elements of the educt state that are not found in the product state are called *educt state's exclusive elements* or, more chemically, the *true educts*. The elements that are found in the product state but not in the educt state are called *product state's exclusive elements* or the *true products*. All elements that are found in both states are the *common elements* or, chemically, the *catalytic elements*.

To convert the overall educt state $\boldsymbol{E}$ completely to the overall product state $\boldsymbol{P}$, all educt state's exclusive elements have to be destroyed and all product state's exclusive elements have to be created.

The progress of an intermediate state used for this part of the heuristic is calculated the following way. First all catalytic elements that were present were removed from the intermediate state $\boldsymbol{I}$ to generate the state $\boldsymbol{I_2}$. Then all true educts that were not present in the state $\boldsymbol{I_2}$ were counted and the number was added to the number of true products found in $\boldsymbol{I_2}$. This number was divided by the sum of the number of true products and true educts to generate a progress value. This progress value was by definition 1 for the product state and 0 for the educt state. It could, however, only assume as many different values as there were true products and true educts.

$$p = \frac{|(\boldsymbol{E} \setminus (\boldsymbol{E} \cap \boldsymbol{P})) \setminus (\boldsymbol{I} \setminus (\boldsymbol{E} \cap \boldsymbol{P}))| + |(\boldsymbol{I} \setminus (\boldsymbol{E} \cap \boldsymbol{P})) \cap (\boldsymbol{P})|}{|(\boldsymbol{E} \setminus \boldsymbol{P}) \cup (\boldsymbol{P} \setminus \boldsymbol{E})|} \tag{4.1}$$

Lemma: The absolute difference between the progress values of two states fulfills the triangular equation and is symmetric. It is, however, only positive semi-definite and thus forms a pseudo-metric.

Proof:

1.)The progress p is a function $f : S \to \mathbb{R}$ where S is a state.

Note to 1.) If the overall educt and product states only contain finite many elements, this function maps infinite many possible states only to finite many values. Thus several states have to be mapped to the same progress value, which is why the difference between the progress values of several states cannot be positive definite. Example calculations show that this is also true for many cases of finite many compatible states.

2.) The absolute distance between two real numbers is a well defined metric on $\mathbb{R}$.

3.) The combination of 1 and 2 makes the absolute difference between the progress values of two states a pseudo-metric on the states.

□

The heuristic demanded that a state should only be further explored if its progress is *greater or equal* the progress value of all its precursors. (Or *smaller or equal*, if search is performed from the product side.)

The calculations with this heuristic showed that this part of the heuristic rarely eliminated states from the search tree that would occur in real reaction path as found in MACiE, or in the shortest path generated by the program. See the results section for more details. The destruction of catalytic elements, on the other hand, was not penalized, although destroyed catalytic elements would have to be created again in order to arrive at the final state. This is due to the fact that the purpose of many catalysts to speed up a reaction by forming intermediates.

4.5.2 Heuristic part 2

The second part of the heuristic uses a different distance measure. We defined our distance d between the two states A and B to be the distance d' between the states $A \setminus B$ and $B \setminus A$. Furthermore we removed all O, H_3O^+ and OH^- molecules from both states, as they are found everywhere and would bias the result.

Definition: Given the underlying distance $d''(a, b)$ between the molecules a and b, the distance d' is defined as follows:

$$d'(A, B) = 1/2 * (\sum_{a \in A} \min_{b \in B} d''(a, b) + \sum_{b \in B} \min_{a \in A} d''(a, b)) \tag{4.2}$$

It remains to be shown whether this distance fulfills the axioms of a metric.

This distance was used due to the following considerations: During chemical reactions substrate molecules are converted to product molecules. Thus it seems natural to find some sort of matching between the molecules of the two states. It would, however, be computationally too expensive to generate all multiset bijections and find the optimal matching. Furthermore this would not even reflect the true situation, as molecules can be split or joined, which is why a 1 to 1 matching might yield wrong results. The first part of the above formula matches each molecule from one multiset to the optimal partner in the other multiset. It is possible that several molecules from one multiset match the same molecule from the other multiset, which reflects the true situation. Unfortunately it is also possible that some molecules from the second multiset are not matched by any molecule from the first multiset. To compensate for this, in the second part of the formula the roles of the two multisets are reversed and every molecule from the second multiset has to match a molecule.

While this formula is not optimal, it is the best compromise between fast computation time and good results that we could achieve. As the results and discussion sections show, this distance can in fact be used for a heuristic that works well in many cases.

As the underlying distance d'' we used the Tanimoto coefficient between the fingerprints of the molecules. These fingerprints and the Tanimoto coefficient were calculated with the openbabel library. We used the FP2 fingerprint model.[94]

While the first part of the heuristic simply discards some states that were not promising, the second part keeps track of the currently best states that should be further explored. See figure 4.1 for an illustration. In the illustration two states would be further explored after each iteration step. Whenever an additional state is generated for a certain search depth, its distance to the product state is calculated. If this distance is better than one of the currently best states, the worst among those best states is flagged and will not be further explored.

The user can specify how many states should be kept at each iteration step. This way the run time of the program does no longer rise exponential with the search depth but about linear.

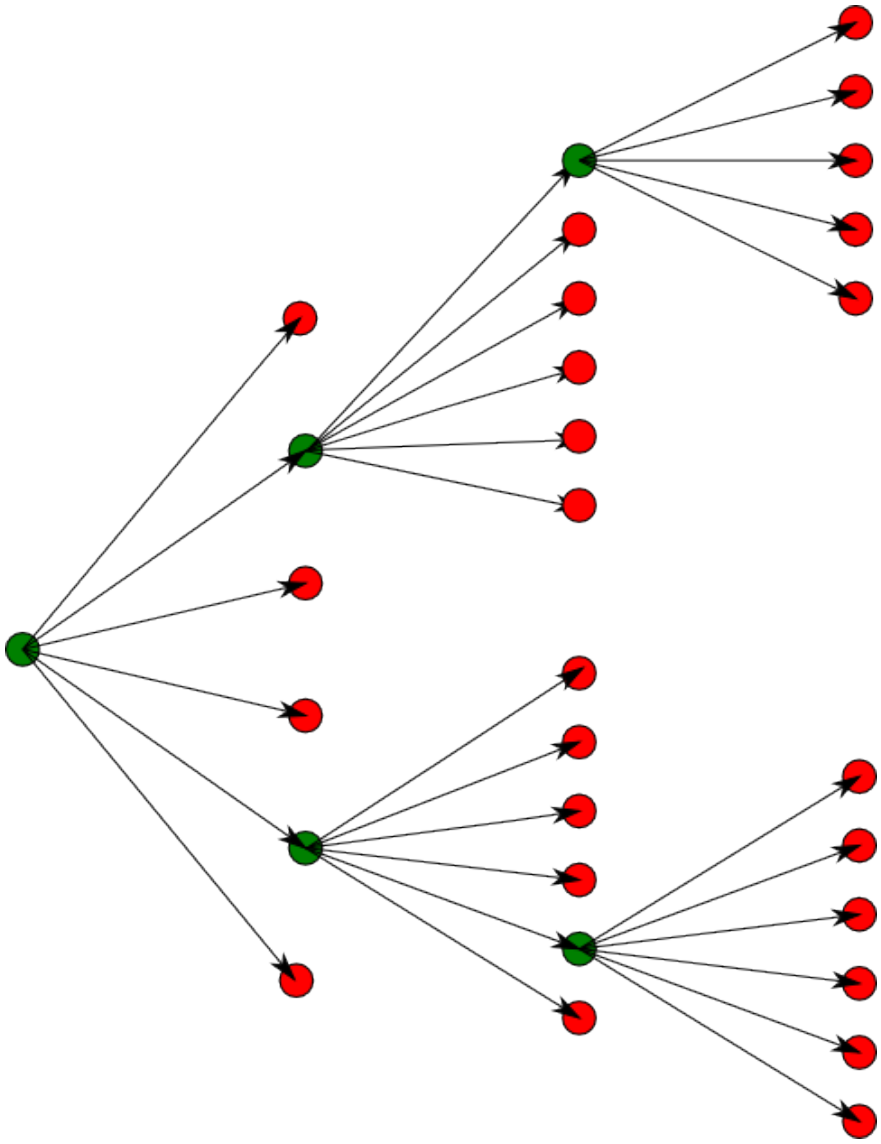


Figure 4.1: After each iteration step, only some states (green) are kept by the heuristic and further explored.

4.6 Additional programs

In addition to MechSearch, a program was written to analyze the output of MechSearch. With this collection of tools it is possible to search for a certain sequence of elementary reactions in the reaction network. This was used to verify whether the reaction path depicted in MACiE was actually found again by MechSearch.

Furthermore, for each reaction rule, the program can write out a representative reaction path from the overall educt state to the overall product state that contains this reaction rule. This representative path is calculated in two parts. First of all it consists of the shortest path from one vertex of the given edge to either overall educt state or overall product state. The second part of this representative path is a path from the other side of the edge to the overall educt or product state, whichever one was not chosen in the first part. For this path, the shortest possible one not containing any edge that was already present in the first part of the path is chosen.

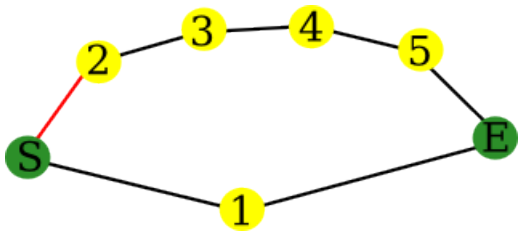


Figure 4.2: If a representative path that contains the red edge from vertex S to vertex 2 is searched for, the program will find the path S-2-3-4-5-E, altough the path S-2-S-1-E (containing the edge S-2 twice) would be shorter.

Part III

Results and discussion

Chapter 5

Evaluation of MechSearch

5.1 Re-finding overall reactions from MACiE

As a first proof of concept, MechSearch was given only the elementary reactions from one single MACiE number, i.e. the elementary reactions corresponding to one single overall reaction. As educts, all molecules that occurred at the substrate side of any elementary reaction but were not produced by any of the previous reaction steps were supplied. As products, all molecules that were found on the right side of an elementary reaction and were not consumed in a later reaction step were used. Then a bidirectional search was performed.

This procedure was applied to all overall reactions from the MACiE database that had an id number smaller than M0100 and for which all elementary reactions were successfully extracted and converted to GML. There were only two reasons why the overall reaction path could not be found: First, some overall reactions in MACiE are stereo isomerizations. As stereoisomers currently cannot be represented in the GGL, these isomerizations are identity reactions in our framework, *i.e.* educt and product state are equal. Whenever educt and product state are equal, MechSearch has no reason to calculate any reaction path and exits right away.

The second problem was related to the aromaticity perception: There were two cases (M0066 and M0084) where the subgraph pattern of the rule was aromaticity corrected in a different way than the molecule it should be applied to. Thus rule application failed. This is a problem that will require further work and probably rethinking of the aromaticity correction procedure. (See also outlook section).

5.1.1 Wrong identity reactions

Furthermore these first calculations showed another complication: Some reactions, like the proton transfer from one residue to another, are identity reactions in our model, as no information about the position of individual residues is present. The program, however, does not store identity reactions in the output, as this would unnecessarily increase the output and the number of possible reaction paths. Thus only shorter paths than MACiE's path are found. This can, however, be overcome by assigning different class labels to an unreactive backbone C of the residues. Now the shorter path is still generated, as the residues can still be used interchangeably for every reaction. The original path from MACiE, however, is now found as well. With this procedure the correct reaction path, as found in MACiE, really was found for all cases where this problem occurred. It is, however, not feasible to assign different labels to identical amino acid residues for calculations with more rules, as this would increase the combinatorial explosion a lot while yielding little benefit.

5.2 Comparison between toyChem and MechSearch

Please note that all calculation times given in this section were generated in real world applications and may depend on different factors. They are given here to give a rough idea of calculation times only.

The toyChem program is part of the GGL. It is used to iteratively apply a set of rules to a set of molecules. The rules are applied to all combinations of molecules. Therefor there is a huge combinatorial explosion when the iteration depth is large.

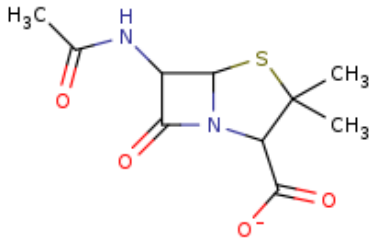


Figure 5.1: A beta-lactam.

The substrate of M0016, a beta-Lactam with functional groups as depicted in figure 5.1, was given to toyChem together with water, OH^-, H_3O^+ and six amino acid side chains. The amino acid side chains were positively charged Lys, negatively charged Asp and uncharged Ser, Cys, Tyr and His. All rules from Ruleset50 were given in forward and backward direction. "Reactions" was selected as an output mode to retain the complete information about the reactions. Note, however, that toyChem is significantly faster if only the SMILES of the new molecules are desired as output. With SMILES as only output, toyChem is faster than MechSearch for short iteration depths.

Depth	Time toyChem	Time MechSearch	# molecules produced toyChem	# molecules produced MechSearch	# reactions toyChem	# reactions MechSearch
1	2 sec	2 sec	11	11	55	40
2	1 min	50 sec	43	38	556	312
3	27 min	16 min	487	121	10256	2370
4	aborted after days	3 h 30 min	-	425	-	16563

Table 5.1: Comparison between toyChem and MechSearch.

The forth iteration of toyChem was aborted when it had not finished after three days.

MechSearch was run with the same input molecules and the same rules but without any heuristic. A Monodirectional search only from the educt side of the reaction was performed. As expected, it also produced 11 new molecules in the

first iteration. In the later iterations – again as expected – less molecules were produced than with toyChem. Table 5.1 gives an overview over the comparison between toyChem and MechSearch. Note that the number of reactions is not equal to the number of edges in the reaction network, as there are cases where the program performs several reactions that correspond to the same edge between two states. See section 3.7.1 for more information.

After four iterations the reaction path from M0016 was generated among others.

The main reason for the speed advantage of MechSearch is that less reactions have to be considered, as reactions between more instances of a molecule than are present at the current time in the active center are not considered in MechSearch. However, the same reaction can often be performed with different states as educts. Then pre-saved reactions can be applied very quickly. Table 5.2 gives an overview over the relation between pre-saved reactions and reactions that have to be performed by subgraph matching.

Iterations	Molecules	New reactions	Pre-saved reactions	Total reactions
1	22	40	0	40
2	49	312	1436	1748
3	130	2370	25908	28278
4	434	16563	287070	303633

Table 5.2: The total number of molecules and reactions generated after n iterations.

Then a calculation from the product side of the overall reaction was done with a depth of 2. It took 5 minutes and generated 76 molecules in total (including 11 molecules given at the start). This demonstrates that the size of the problem very much depends on the substrate molecules (from the educt side two iterations only took 50 seconds), as the program will take longer if many reactions can be performed with the substrate.

As can be seen clearly from those numbers, the approach of MechSearch especially pays off for larger iteration depths. This significant increase in speed for larger iteration depths is due to the reduction of the search space as many combinations of molecules can not concurrently occur in the same state.

When MechSearch was run with a depth of 2 in a bidirectional fashion, it generated 95 molecules, performed 955 reactions by subgraph matching (and 3531 pre-saved reactions) and took 6 minutes in total. Again the reaction path from MACiE for M0016 was among the generated paths.

5.3 Exhaustive calculations of some reactions

As can be seen from the above example, with Ruleset50 calculations with an iteration depth of 3 can be easily done and calculations with a depth of 4 are still possible. Therefor some reactions with an overall number of 5 to 7 steps were exhaustively calculated in a bidirectional way with this ruleset.

Exhaustive calculations were performed for the overall reactions of MACiE M0002, M0005, M0006, M0025, M0030, M0031 and M0049. For each overall reaction three calculations were performed, termed "norm", "aa" and "cof": At first only the substrate and product molecules plus those amino acids and cofactors that react in the mechanism displayed by MACiE were provided as educt and product state. For the second calculation, more potential reactions were allowed by adding the remaining of the six amino acids Ser, Glu, Cys, Lys, His and Tyr each at least in two copies. If they were present for the reaction in a different protonation state, the protonation state from MACiE was adopted for both copies. For the third calculation a different type of chemistry was allowed by adding one copy of the cofactors NAD+, NADH and FAD, if they were not yet present.

In the case of M0049, the C-terminal carboxylic acid group of Ser reacts. This was replaced by a Glu side chain for our calculations.

5.3.1 M0002 beta-lactamase (Class A)

The overall reaction with the id M0002 is the hydrolysis of a beta-Lactam to a substituted beta-amino acid by the beta-lactamase (EC number 3.5.2.6) class A found in *E. coli*. The reaction mechanism depicted in MACiE consists of 5 steps (figure 5.4).

Exhaustive calculations with MechSearch ("norm") took 20 min. Together with additional amino acids ("aa") it took 1 h 51 min. and with additional cofactors ("cof") it took 3 h 51 min. As expected, the path depicted in MACiE was – among other paths – found in all three calculations.

The shortest path found by MechSearch was also present in MACiE with the MACiE id of M0015. This is the di-zinc dependent mechanism of class B beta-lactamase as found in *Bacteroides fragilis*. As an exhaustive calculation was performed, the shortest path found is the shortest reaction path possible with the given starting material and the chemistry present in Ruleset50.

MACiE entry M0016 is the mono-zinc dependent mechanism of beta-lactamase class B. The corresponding reaction path was also found in the output of all three calculations.

Calculating a representative path for all reaction rules yields some notable paths from educt state to product state. Beside the paths as depicted in MACiE for M0002, M0015 and M0016 there is one more possible reaction worth mentioning that does not need additional molecules. Step 3 of M0029 is the addition of water to a carbonyl C=O double bond. This can replace step 4 of M0002 if an additional proton transfer is performed. In the presence of additional amino acids, the ester hydrolysis described by steps 4 and 5 of M0002 can be achieved differently as well: Steps 3 and 4 of M0029 catalyze this transform using a Tyr-residue and steps 4 and 5 of M00b05 use a His for this hydrolysis. No additional interesting paths were found in the presence of cofactors.

The energy diagram 5.3 for this reaction reveals the main problem with energy as a criterion to distinguish between several paths. The paths for M0002, M0015 and M0016 are all found in nature. The energy diagram, however, gives the impression that M0015 would be preferable over the other reactions because it is the path that corresponds to the lowest energy. Furthermore the highest energy difference covered by one step in this path is lower than that of the other reaction pathways. However, this picture does not show all necessary information because the activation energy could not be calculated. The local maxima in path M0002 are due to tetrahedral carbonyl addition intermediates. For reaction M0015 no such intermediate is depicted in MACiE or one of the two references[95] given by MACiE. This could be due to two reasons. On the one hand it could be an inaccuracy in the paper. On the other hand it is possible that inside the enzyme no tetrahedral intermediate is formed, but the reaction proceeds through a tetrahedral *transition state*. While it is generally accepted that nucleophilic substitutions proceed via a two step addition-elimination mechanism in aqueous solution, there are studies that suggest a one-step mechanism with a transition state in gas phase[96–98]. As enzymatic reactions sometimes resemble gas-phase reactions more than reactions in solution, this could be true for enzymes as well. In that case the depicted energies in the diagram are correct but there is a high activation energy to be assumed. No matter how the reaction proceeds in nature, the energy to form the tetrahedral species is needed for the reaction to proceed, no matter if it is an activation energy to a transition state or a reaction energy to a (reactive) intermediate.

For the program MechSearch this means that the energy calculated for the individual states is no good criterion to pick the correct path among several good paths as long as it is not possible to estimate the activation energy in a quick and accurate way.

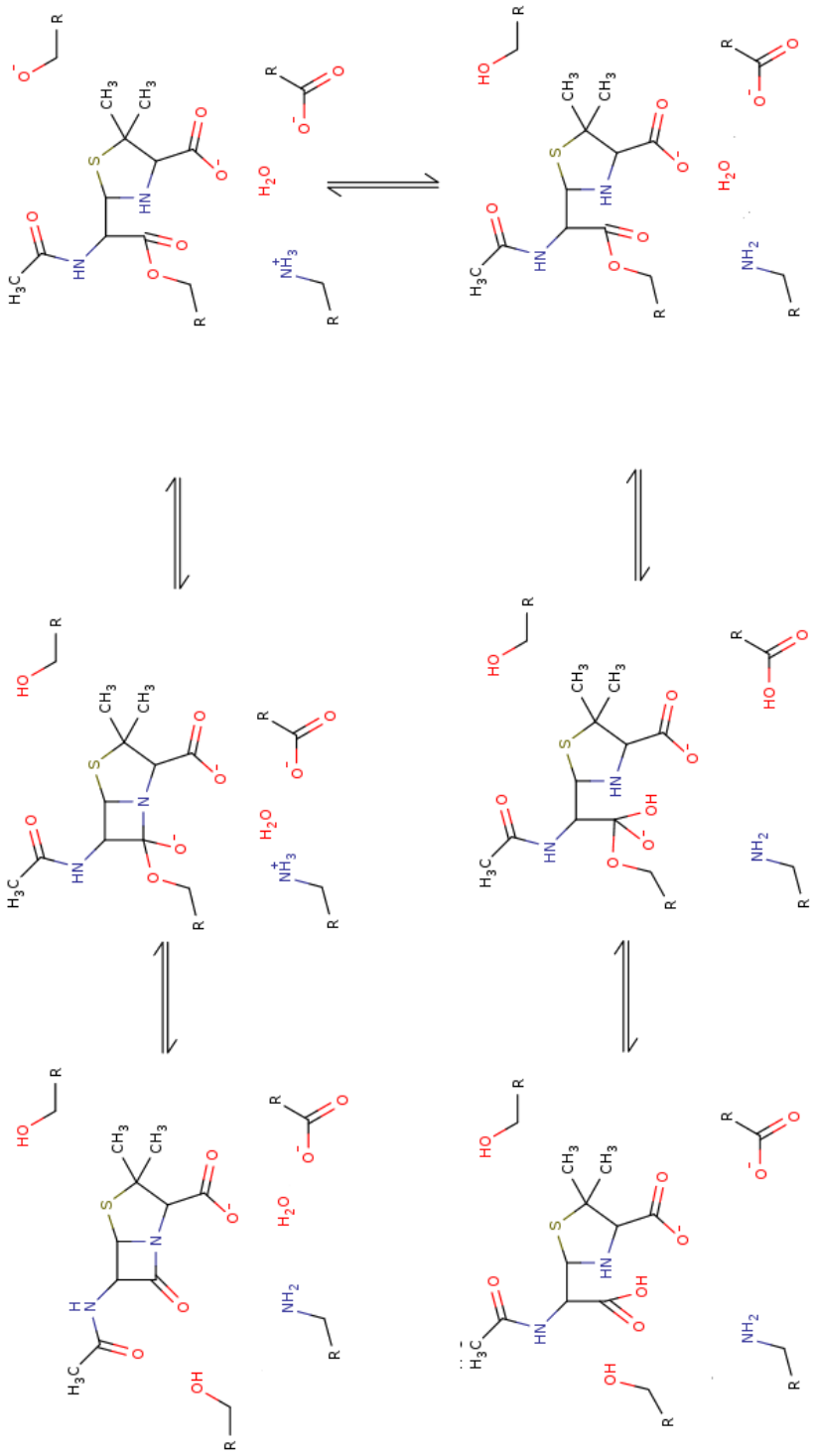


Figure 5.2: Reaction Mechanism of M0002.

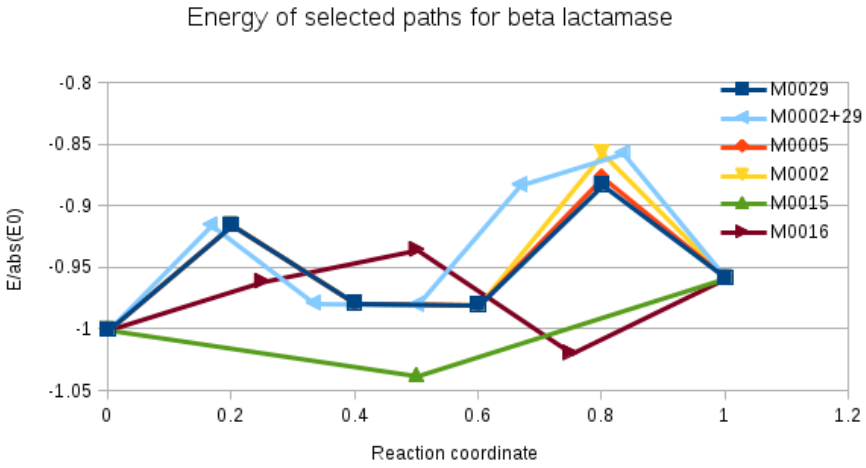


Figure 5.3: The energy of some reaction paths found by exhaustive calculation of M0002 with additional amino acids.

5.3.2 M0005 - Carboxypeptidase D

The reaction with MACiE number M0005 is catalyzed by carboxypeptidase D (EC 3.4.16.6) from wheat (*Triticum aestivum*). At low pH this enzyme specifically removes basic or acid residues from the C-terminus of a peptide.

After an initiating proton transfer from serine to histidine the alcoholate attacks the carbonyl carbon to form a tetrahedral intermediate. This intermediate collapses in the next step and eliminates the amino acid. The last two steps are the hydrolysis of the serine-peptide bond, again via a tetrahedral intermediate.

The exhaustive calculation took 15 sec. with no additional amino acids, 5 min. with amino acids and 17 min. with the cofactors added.

Instead of M0005.stg04 the reaction with number M0029.stg03 can be used. With additional amino acids, M0029 steps 3 and 4 can be used instead of M0005 steps 4 and 5.

Altogether M0005 is somehow similar to M0002, since addition-elimination mechanisms to a C=O double bond are involved in both cases.

5.3.3 M0006 - Glutathione-disulfide Reductase

The overall reaction with the MACiE number M0006 is a redox reaction. Thus a different chemistry than in the cases of M0002 and M0005 can be expected. M0006 is the reduction of glutathione by oxidation of NADPH catalyzed by human glutathione-disulfide reductase (EC 1.8.1.7).

Exhaustive calculation of M0006 took 23 sec. if no additional molecules were added and 6 min. with additional amino acids. Note that the cofactors NADPH and FAD are part of the reaction and thus present in all calculations.

This enzyme possesses an intramolecular disulfide bond. In the course of the reaction (figure 5.4), NADPH is used to cleave the intramolecular disulfide bond first, which in turn cleaves the disulfide bond of glutathione. Thus the correct path can only be found if the disulfide bond of the substrate and the disulfide bond of the cysteine residues are labeled differently. However, shorter paths will be found as well, as within our framework non of the reasons why intermediate reduction of the intramolecular disulfide bond is favorable can be represented. This problem was also described in section 5.1.1.

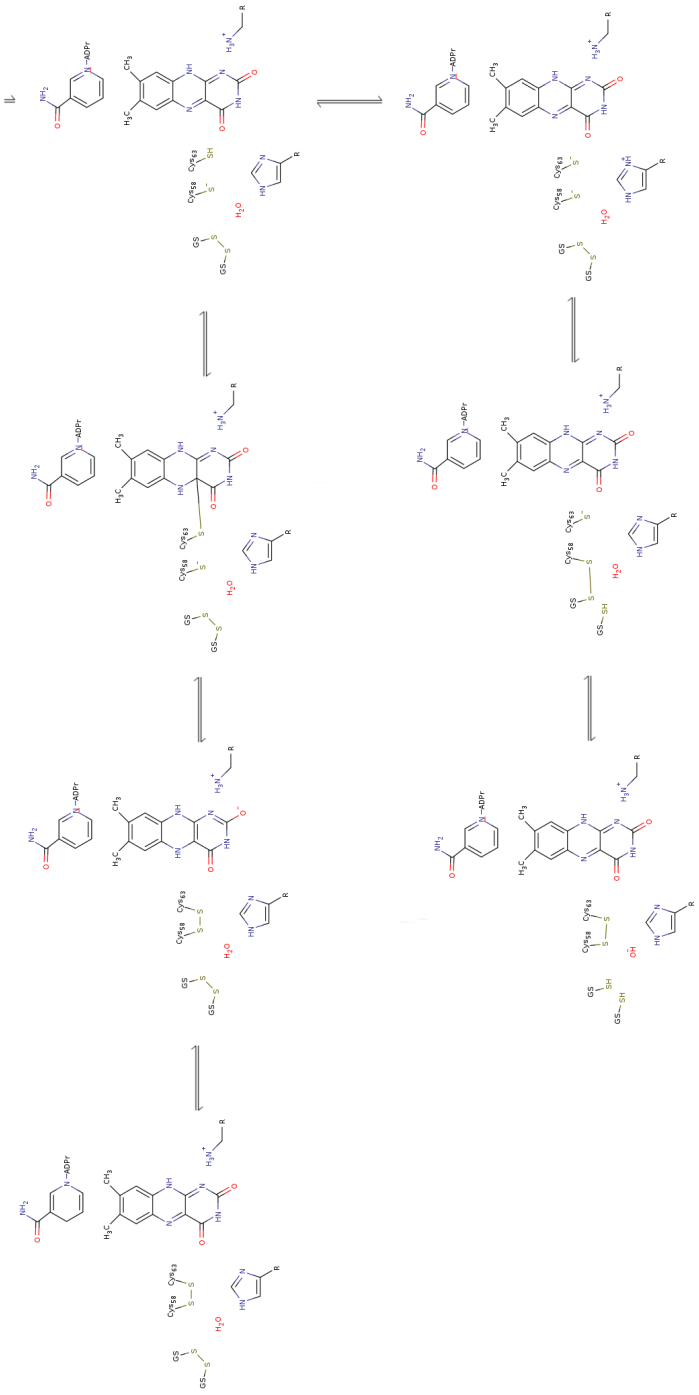


Figure 5.4: Reaction Mechanism of M0006. Right now the program MechSearch has no criterion to distinguish the Cys-Cys disulfide bond from the disulfide bond in Glutathione.

5.3.4 M0025 - N-Carbamoylsarcosine Amidase

The reaction with MACiE number M0025 is catalyzed by N-carbamoylsarcosine amidase from *Arthrobacter sp.* with the EC number 3.5.1.59. It proceeds in 5 steps. In the first step sulfur from Cys attacks the amide carbon to form a tetrahedral intermediate, which in the second step eliminates NH_3. In the third step water attacks the same carbon to form another intermediate, which in the forth step eliminates Cys again. The product finally decarboxylates in a non enzymatic reaction.

Exhaustive calculation for this reaction took 20 sec. With additional amino acids it took 6 min. and with cofactors it took 17 min. Next to the correct path found in nature no notable paths were found by these calculations. This is probably due to the fact that Ruleset50 is too small, as several other mechanisms would also be chemically plausible: The amide C could be attacked by oxygen residues as well and could potentially also eliminate the other nitrogen first.

5.3.5 M0030 - C-Acetyltransferase

Formate C-acetyltransferase (EC 2.3.1.54) from *E. coli* catalyses the reaction of pyruvate with CoA to acetyl-CoA and formate by a radical mechanism. The enzyme contains a glycyl radical which is used to start the radical reaction mechanism and is restored after the reaction.

Here calculation took 16 sec., 42 min. and 6 h 23 min. for the normal case, the calculation with additional amino acids and the calculation with cofactors, respectively.

Beside the path depicted in MACiE a shorter reaction path was found that used less radical transfers between different sulfur atoms. The energetic and steric reasons why these additional radical transfers are favorabe in nature currently cannot be modeled within our framework.

5.3.6 M0031 - Thymidylate Synthase

Thymidylate synthase (EC 2.1.1.45) from *Lactobacillus casei* catalyses the methylation of dUMP to 5-methyl dUMP by the use of (6R)-5,10-methylene-tetrahydrofolate, which is converted to dihydrofolate.

Exhaustive calculations with MechSearch took 11 min. Together with additional amino acids it took 10 h 20 min. and with additional cofactors it took 18 h 30 min. Despite variants of the reaction path shown in MACiE no other paths were found by this calculation.

5.3.7 M0049 - Histidine Decarboxylase

Histidine decarboxylase (EC 4.1.1.22) from *Lactobacillus 30a* catalyses the decarboxylation of histidine to histamine.

Exhaustive calculations with MechSearch took 12 sec. Together with additional amino acids it took 7 h and with additional cofactors it took 21 h 30 min.

In the first step the terminal carbonyl carbon of the PTM pyruvoyl residue is attacked by the substrate's amine group. Then a Schiff base is formed and water is eliminated. Step three is the decarboxylation of the substrate together with a double bond rearrangement. Steps 4, 5 and 6 describe the reverse double

bond rearrangement and the hydrolysis of the Schiff base. Step 7 is a proton transfer to restore the original enzymatic state.

Again the only notable path found was the one from MACiE.

5.4 The search for a heuristic

In the search for a heuristic some molecular properties were investigated.

5.4.1 The Jankowski energy

One of the first ideas was to sum up the Jankowski energies of all molecules and residues of a state. The idea to use this energy value as a criterion to differentiate between correct and incorrect reaction paths was dropped because of several reasons. First of all the Jankowski energy cannot be calculated for all types of molecules right now because fragments with high energy such as C=[O+]-C and radicals do not yet have a tabulated energy value. Secondly, as illustrated in 5.3.1, the energy of transition states is often more important than the energy of intermediates, as the transition state energy determines the activation energy. Unfortunately the transition state energy can not be estimated in a quick and accurate way.

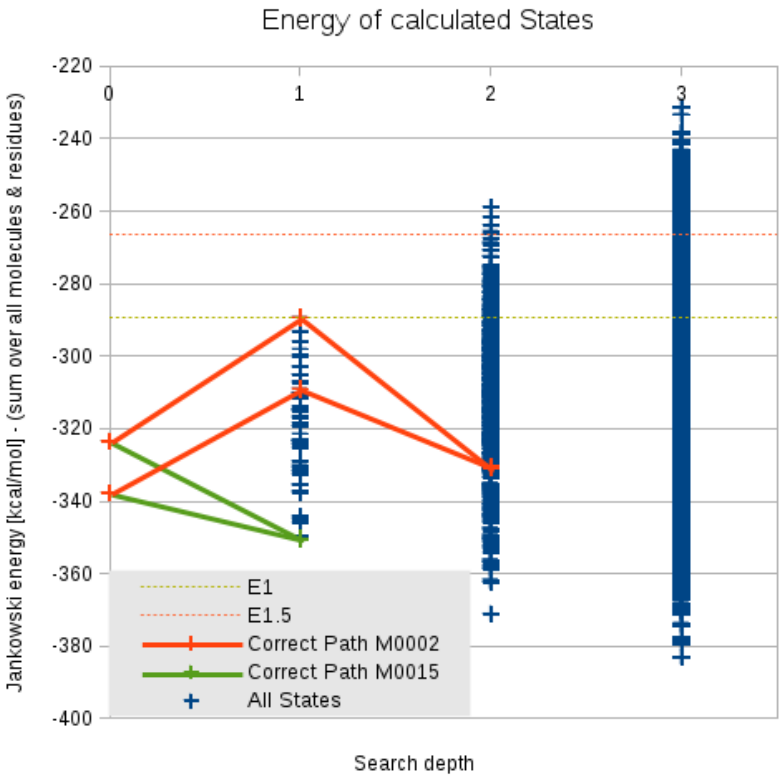


Figure 5.5: The energy distribution of all states generated with MechSearch for M0002 (blue). The reaction paths found in M0002 and M0015 are shown in red and yellow respectively. Note that there are two states with an iteration depth of 2 and a similar energy on the path of M0002, one from each side of the bidirectional search.

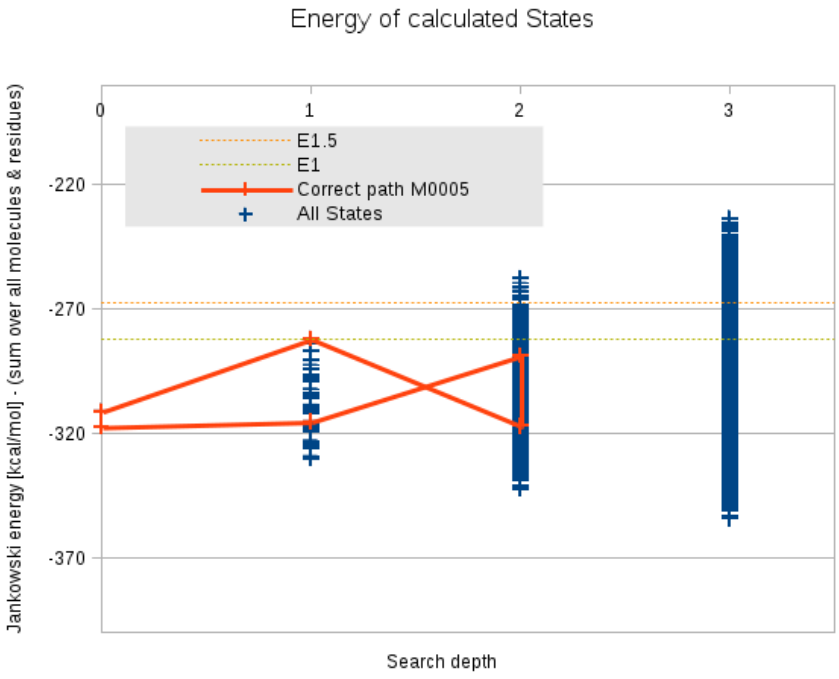


Figure 5.6: The energy distribution of all states generated with MechSearch for M0005.

One idea was to use the Jankowski energy at least for discarding intermediate States with a unrealisticly high energy. However, as figure 5.5 illustrates, in the second and third iteration steps only very few States actually have an unrealistic energy value. This is due to the fact that the Jankowski energy is mostly based on molecular fragments and the graph grammar rules used in our algebraic chemistry only modify fragments of molecules. Thus, if additional energy terms for strain and aromaticity are neglected, one can associate a difference in energy with the rules and not only with the actual reactions. As all rules were taken from reactions that occur in nature, no energy difference of a single rule, and thus of a single reaction, can be unrealistically high.

Therefor all states have realistic energy values after the first iteration. In fact the states that correspond to the correct reaction path from M0002 and M0015 are among the states with the highest and lowest energy, respectively. Only in the second iteration, applying a reaction rule that will increase the energy to a state with a high energy could potentially be forbidden by a heuristic. As seen from figure 5.5, this would reduce the amount of states after two iterations only by a very small percentage. Only in the third iteration step such a heuristic could have significant effects.

In the diagrams 5.5 and 5.6 two horizontal lines are present. One is at the energy of the highest State after the first iteration (E_1). However, using this in a heuristic would probably be too restrictive. The second line (at $E_{1.5}$) could potentially be used to discard all States with higher energy. It is calculated by

adding 1.5 times the highest energy difference encountered in the first iteration step to the maximum of the energy of the overall educt state and the overall product state.

Table 5.3 gives the percentage of states above those two cutoff lines for some exhaustively calculated examples with Ruleset50 and additional amino acids (see 5.3). It can be seen that after two iterations only a very small percentage of states is over the higher cutoff line. Only in the third iteration step about 7% of all states could potentially be discarded. If the heuristic estimates the energy before rule application and does not even create these states, this could potentially lead to a noticeable but not really great reduction of calculation time for multistep reactions – at least if it was possible to calculate the energy for all molecules.

As we currently mainly deal with shorter iteration depths and as the energy value cannot yet be calculated for all molecules, this type of heuristic was not yet implemented but remains as a task for the future.

Reaction	iteration depth = 2		iteration depth = 3	
	above E_1	above $E_{1.5}$	above E_1	above $E_{1.5}$
M0002	12%	0.6%	33%	6%
M0005	15%	1.5%	40%	15%
M0006	8%	1%	20%	6%
M0025	5%	0.5%	20%	5%
M0031	7%	1%	19%	7%
M0049	10%	0.7%	42%	23%

Table 5.3: The total number of molecules and reactions generated after n iterations. As M0030 involves radicals, no energy values could be calculated for that reaction.

5.4.2 Complexity

Bertz *et al.* distinguish two types of complexity[69]: intrinsic complexity and extrinsic complexity. Extrinsic complexity is an estimate for the synthetic accessibility and thus depends on the symmetry, while intrinsic complexity only depends on the constituent parts of the molecule itself.

Bertz uses complexity versus step plots to evaluate the complexity of synthetic routes.[69] He defines the excess complexity as the area under the complexity versus step plot of the synthetic route minus the area of a one step transform of the educts to the products.

Complexity was calculated for all molecules except for some aromatic molecules where treating aromatic bonds as a bond order of 1.5 failed.

Unfortunately calculating the excess complexity in a similar fashion for a reaction mechanism did not serve as a good heuristic. The reason is the following: While the synthetic complexity certainly increases with additional synthetic steps, this is not necessarily the case when it comes to elementary reactions. On the contrary, one aspect of catalysis is facilitating a reaction by allowing for an alternative reaction route that often consists of more steps. However, additional elementary reaction steps in general do not reduce the overall reaction complexity either. Calculating an average over all steps is thus not possible

because in that case additional proton transfer reactions that have nothing to do with the desired transform would reduce the complexity.

Furthermore the same problem that already applied to energy also applies to complexity: As we only know the complexity of the intermediates but not of the transition states, an important part of the information is missing.

5.5 The final heuristic

Finally, a heuristic consisting of two parts, as described in the methods section 4.5, was used. With this heuristic the program's performance and quality of solutions were evaluated.

5.5.1 Illustration of the distance used for the heuristic

The overall reaction with MACiE id M0036 is a very good example to illustrate how our heuristic works. If this reaction is calculated with a bound on the search breadth of 1, no reaction path is found, as the heuristic eliminates the correct intermediate state. With a bound on the search breadth of two, however, a reaction path is found.

After the first iteration from the educt side 56 intermediate states are generated. At the same time 62 states are generated from the overall product state, as we use a bidirectional search approach. For each of the 56 intermediate states generated from the educt side the distance to the overall product state is calculated. Then the intermediate states are ordered according to their distance and only those states with the smallest distance to the overall product state are further explored. The same procedure is applied to the states that were generated from the overall product state, but here of course the distance to the overall educt state is used.

If a reaction path of length 3 exists, it is enough if a correct state is selected by the heuristic either on the educt side or on the product side. If only paths of length 4 or longer exists, correct decision on both sides is necessary.

On each side the heuristic selects the n states with the smallest distance for further exploration.

Figure 5.7 shows the two intermediate states generated from the overall educt state that have the best distance value. If a bound on the search breadth of 1 is chosen by the user, only the state with the smallest distance will be further explored. In this case no reaction path will then be found. With a bound on the search breadth of 2 or more, however, the second best state is also selected, which in this case will allow a reaction to one of the states generated from the overall product state (not shown in the figure).

Figure 5.7 also illustrates how the distance is calculated. Only the molecules that are not found in both states but would actually have to react, are taken into account, except for water in all protonation states, which never contributes to the distance. The figure shows an interesting aspect of our distance measure: Often a one to one mapping between the molecules from two states is not possible, especially if the number of molecules differs between the two states.

If one molecule is split up into two parts, the Tanimoto coefficient between the original molecule and each of the fragments will probably be rather high. This is compensated by the fact that the overall distance is a sum of only three contributions. If, on the other hand, two molecules undergo a reaction that modifies both reagents only a little and yields two products, the individual Tanimoto coefficients will probably be smaller but the total distance will be a sum of four contributions.

One downside of the Tanimoto coefficient is, however, the fact that two larger molecules often have a higher similarity than two smaller molecules. This is due to the fact that only the bits that are set in both fingerprints contribute to

similarity, while bits that are 0 in both fingerprints are not explicitly counted. On the other hand, this has the advantage that fingerprints can be arbitrarily long and can consist mainly of zeros, for example ClOH for example, which consists of only 2 fragments, Cl and ClO (O is ignored because it is too common). Cl^- obviously consists of only 1 fragment, Cl. The Tanimoto coefficient between these two molecules would then be 1/2 because 1 out of two total fragments is common in both molecules.

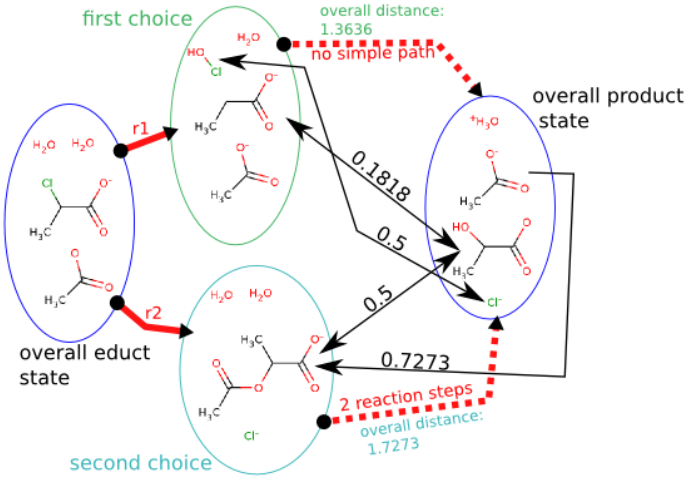


Figure 5.7: Calculation of the distances of the two best States generated from the overall educt state (blue). Note that the reactions corresponding to the dashed red lines are not known at the time of calculation of these distances.

5.5.2 The optimal bound on the search breadth

With the final heuristic, after each iteration a certain number of states are chosen for further exploration, while all the others are marked as uninteresting. The value of how many states should be kept after each iteration, the bound on the search breadth, can be set by a user flag.

All overall reactions from MACiE with an id smaller than 100 for which all elementary reactions were part of Ruleset100 and Ruleset250 were used. Thus, as shown in the proof of concept section, the correct reaction path should always be found if an exhaustive calculation was performed. Each calculation was done twice, once only with the educts and products that took part in the reaction mechanism and once with additional amino acids (the same amino acids as described in section 5.3). As a search depth the minimal number necessary to find the correct reaction path was used.

In figures 5.9, 5.10 and 5.11 the percentage of reactions where a reaction path was found with the heuristic is plotted. Results for reactions with less than 3 steps are not plotted because these reactions are always exhaustively calculated and thus a path is always found. This is due to the fact that in the first iteration all rules are applied to the overall educt and product states and the selection of states to be further explored is performed afterwards. Thus paths of a length

smaller than 3 are always found after the first iteration before the heuristic takes effekt.

The reason I plotted how often a reaction path was found and not how often the certain path depicted in MACiE for that reaction was found was the following: First of all, in many cases permutations of the order in which the elementary reactions occur are possible. The probability to pick exactly the one path from MACiE is then reduced dramatically, while for most applications it does not matter in what order two reactions are applied. Furthermore there are cases, like the overall reaction of M0002, where several reaction mechanisms are possible and even found in MACiE. The more correct reaction mechanisms exist, the smaller the probability to find a specific one, even if the heuristic works perfectly well. As most of the shortest paths found by MechSearch are chemically equally good, I just counted the number of reactions where a path was indeed found with the heuristic.

For the figures 5.9 to 5.11 all reactions where a path shorter than the path depicted in MACiE existed were discarded. From the remaining calculations for each length the percentage of calculations where at least one reaction path was found was then calculated and plotted in these figures. That left a total of 9 reactions consisting of 3 steps, 17 reactions consisting of 4 steps and 6 reactions consisting of 5 or 6 steps. For paths of length 3 and 4 an estimated value for an imaginary heuristic that randomly picks a given number of states was plotted as well. This random value was calculated as follows:

For paths of length 3 it was noted that the correct State for further exploration had to be chosen either from the states created from the overall educt state or from the states created from the overall product states or from both (see figure 5.8). Thus, for each of the calculations where a path of length 3 existed, the number of states created on each side and the number of different possible reaction paths were taken from the exhaustive calculation of this example. With these values the probability of a random heuristic picking one of the correct states on either side was calculated and the average of these probabilities was plotted.

For paths of length 4 the probability for a random heuristic was calculated analytically from the hypergeometric distribution. The average number of states created after the first iteration with Ruleset100 was 65 states. It was assumed that 3 states on each side were part of a potential reaction path and that any combination of these 3 states from each side was possible. Thus the probability of choosing at least one out of 3 correct states from a total of 65 states was squared to give the total probability.

For reactions consisting of 4 or 5 steps no value for a random approach was calculated, as the probability in this case depends on too many factors that could not reliably be estimated, like the number of similar paths.

The probability to find a reaction path is 1 for paths of length 3 or 4 if all states created after the first iteration are further explored. This is due to the fact that whenever the bound on the search breadth is larger than the number of states generated in the first iteration, calculations become exhaustive for the first 4 steps. Thus the plot of the probability for the random heuristic has a sigmoid shape on the logarithmic scale and approaches 1 if enough states are explored. In the left part of the diagram, however, the heuristic based on the distance measure as defined in section 4.5.2 is significantly better than random. Indeed in almost one third of all cases it is enough to select one state at each

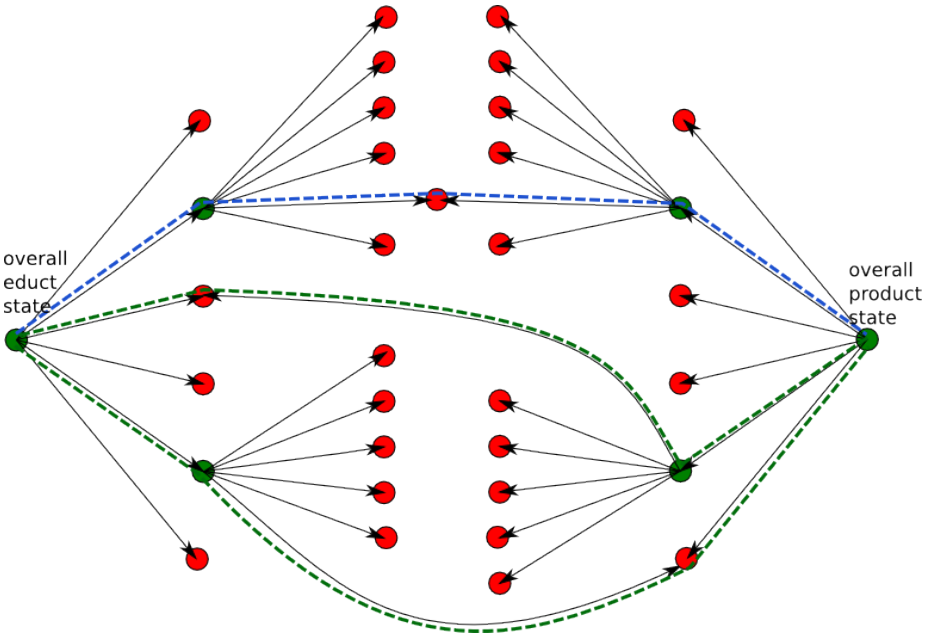


Figure 5.8: Search with an iteration depth of 2 per side. Green states are selected for further exploration by the heuristic. For a path of length 3 the correct state has to be selected at either side of the search tree (green dashed lines). For a path of length 4 the correct state has to be selected at both sides (blue dashed line).

side, and for about 50% of all cases the heuristic brings a significant advantage.

The fact that the random heuristic seems to be better than the heuristic based on the distance if calculation is close to exhaustiveness could simply be an artifact from the small sample size. It could, however, also mean that reactions for which a mechanism could not be found with a smaller bound on the search breadth are activated in a way that increases the distance to the product state. In that case, a high number of proton transfers that do not change the distance would be selected for further exploration instead.

It would therefor be desirable to somehow sum up all states that only differ by a single proton transfer as one superstate. Indeed some preliminary calculations were performed with such a superstate approach. The idea, however, was dropped because generating all possible protonation states for each molecule and trying each of them on every rule seemed computationally too expensive. This was due to the generation of some multiple (de-)protonated molecules that would not have been generated during normal calculation. Furthermore deprotonation of nitrogen atoms could lead to a change in aromaticity, a step that could not simply be reversed, as illustrated in section 3.8.1. It might, however, be an option to retry this approach together with an adequate heuristic.

As a conclusion one can say that in about 50% of all cases the heuristic brings a dramatic advantage over exhaustive calculation for reaction paths consisting of a small number of steps. For larger numbers of steps exhaustive calculation is not feasible and using a heuristic that yields a result in about 50% of the cases is certainly better than not being able to calculate such cases at all.

One interesting aspect is the fact that for 3 step reactions the success rate was higher with Ruleset250 than with Ruleset100, although a solution existed also for Ruleset100 alone. This could be due to the fact that more rules also allow for more reaction paths. As our sample size is rather small, it may, however, as well be by chance. The situation is similar for calculations with additional amino acids. These additional amino acids may allow for a new reaction path and thus increase the success-rate, but they also increase the number of states, especially of states with similar distance, and thus make it harder to find a given path.

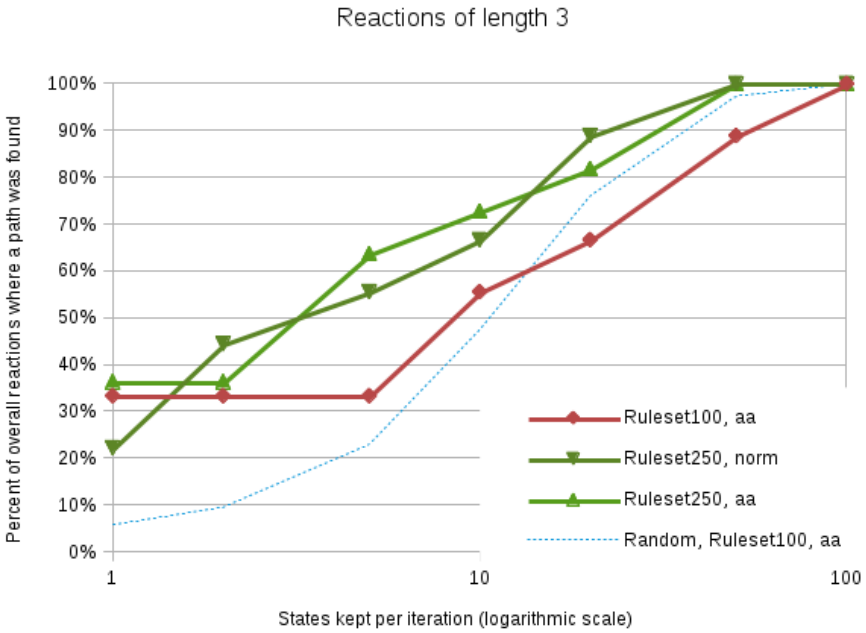


Figure 5.9: % of reactions where a reaction path was found when calculated with a maximal depth of 2 per side and a reaction path of length 3 existed. Note that the dashed line for random choice only applies to Ruleset100. As Ruleset250 contains more rules, it also generates more states per iteration.

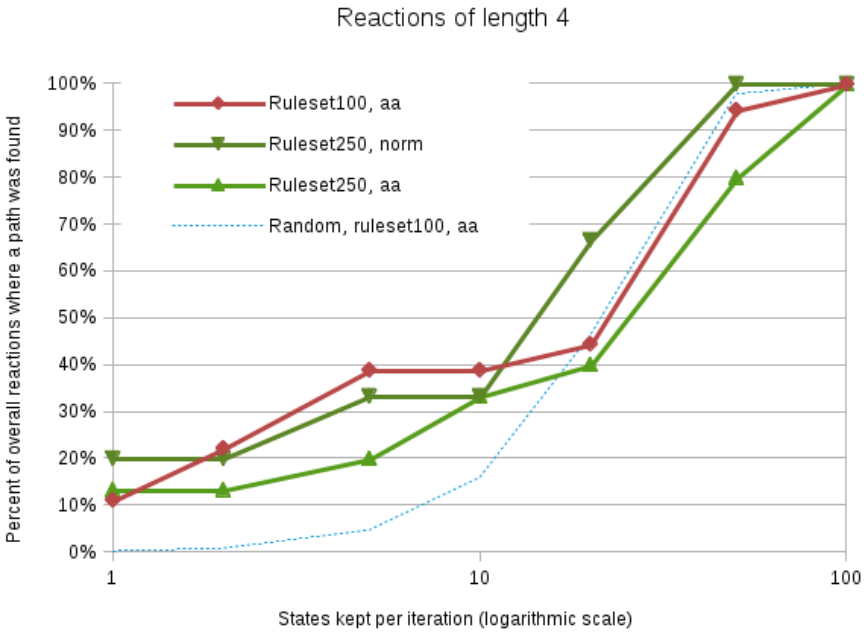


Figure 5.10: % of reactions where a reaction path was found when calculated with a maximal depth of 2 per side and a reaction path of length 4 existed. Note that the dashed line for random choice only applies to Ruleset100.

Figure 5.11: % of reactions where a reaction path was found when calculated with a maximal depth of 3 per side and a reaction path of length 5 or 6 existed.

Chapter 6

Application of MechSearch

Although the fact that an enzyme is orphan does not mean the reaction mechanism for this enzyme is unknown, it could mean that this enzyme is not very well studied and the reaction mechanism might be unclear. Furthermore, for applications where metabolic networks with many reactions should be studied it is often not possible to do elaborate literature research to find the correct reaction mechanism for every single reaction. I thus decided to take the seven reactions from ORENZA that correspond to the KEGG[99] pathway "00770 Pantothenate and CoA biosynthesis" without verifying that the reaction mechanism of these reactions is indeed unknown. For these 7 reactions I tried to calculate a reaction mechanism with the program MechSearch using Ruleset250. I added the cofactors and substrates that were mentioned in the KEGG database as well as the usual 6 additional catalytic amino acids.

Indeed, a reaction mechanism was found after the first iteration step for 3 of the 7 reactions, namely the reactions with KEGG id R02474, R02971 (however with deprotonated phosphate as input molecule) and R02471. These reactions are catalyzed by the orphan enzymes EC 3.5.1.22, EC 1.2.1.33 and EC 1.1.1.106, respectively.

EC 3.5.1.22 catalyses the cleavage of an amide bond. MechSearch proposed a mechanism consisting of the two steps used by HIV-1 retropepsin (EC 3.4.23.16, M0175) to cleave peptide bonds. EC 1.2.1.33 catalyzes the phosphorylation of Pantetheine. As phosphorylations are common in nature and easily achieved in a two step mechanism, this reaction was no challenge for the program MechSearch. It proposes the first step of M0035 (phosphorylase kinase, EC 2.7.11.19) – a nucleophilic attack of the phosphor atom – followed by a simple proton transfer. EC 1.1.1.106 oxidizes a primary alcohol to the aldehyde using NAD^+. The key step proposed by MechSearch for this reaction is stage 3 of the mechanism of M0093 (hydroxymethylglutaryl-CoA reductase (NADPH), EC 1.1.1.34) which has to be prepared by a proton transfer to water.

After exhaustive calculation with a search depth of 3, a path was also found for EC 1.2.1.33, (R)-dehydropantoate dehydrogenase. This enzyme oxidizes an aldehyde to a carboxylic acid using NAD+. The only aldehyde dehydrogenase present in MACiE is betaine-aldehyde dehydrogenase (M0100, EC 1.2.1.8). Rules derived from this reaction, however, could not be applied to our substrate because the nitrogen attached to the alpha-carbon of the substrate is part of the rule's context, as carbon atoms attached to hetero atoms count as functional

groups and are included during extension of the reaction core. This behavior of functional group extension is desired this way, as a positively charged alpha substituent may certainly influence the reactivity of the substrate. Altough the rules from betaine-aldehyde dehydrogenase could not be applied to the substrate of (R)-dehydropantoate dehydrogenase, a reaction mechanism was still found by MechSearch. This mechanism uses rules from different EC numbers to achieve this goal. It was achieved by application of the following steps:

1) A cysteine is deprotonated by histedine and attacks the aldehyde group in one step, according to a rule from step 6 of the mechanism of acetyl-CoA C-acyltransferase (M0077, EC 2.3.1.16) or from step 2 of hydroxymethylglutaryl-CoA reductase (NADPH), an enzyme that reduces a thioester to an alcohol (M0093, EC 1.1.1.34).

2) The tetrahedral intermediate eliminates a hydride which is accepted by the NAD+ cofactor. The reaction rule for this step was step 1 of the mechanism of hydroxymethylglutaryl-CoA reductase (NADPH).

3) Further steps from M0093 could not be used, as M0093 generates an alcohol instead of a carboxylic acid. Now a rule from step 5 of M0234 (GMP synthase (glutamine-hydrolysing), EC 6.3.5.2) is used. Water attacks the thioester group to form a negatively charged tetrahedral intermediate, while another histedine is protonated.

4) Now the first step, which was the attack of cysteine on the carbonyl group is reversed, as the tetrahedral intermediate collapses to eliminate cystein, which deprotonates one of the protonated histedines.

5) Finally water deprotonates histidine to restore the enzyme to its original state. In fact a total of three paths of length 5 was found by MechSearch. The other two paths are identical to the one described here except for the fact that the last step of this mechanism was the third or fourth step in the other reaction paths. While a lot of paths with length four or longer were found, none of them introduced any relevant new chemistry, as they all relied on the key steps that are the first two steps of this mechanism.

This reaction shows some characteristics of the program MechSearch. Due to the fact that all rules were extracted from enzymes, the solution is not only chemically meaningful, but has typical features of enzymatic reactions, such as the use of thioesters to activate carbonyl groups. Indeed the mechanism, although not generated by rules from an aldehyde dehydrogenase, is very similar to the mechanism of well studied aldehyde dehydrogenases. One of those examples can be found in MACiE with an id of M0100.

This example also shows the great challenge reaction mechanisms pose to any heuristic. With the distance based heuristic implemented in MechSearch, the following problem arises: After the first step, the enzyme's state has a higher distance to the product than the overall educt state. This is because the C=O fragment of the substrate that is also a substructure of the product's carboxylic acid group has disappeared, while fragments containing a C-S-C substructure were created that are not found in the overall product state. Useless proton transfer reactions, on the other hand, do not change the distance at all and are thus preferred by the heuristic. In the case of this reaction the distance measure we used has another problem: NADH is (wrongly) perceived as aromatic by the GGL's aromaticity perception, just like NAD+. As the fingerprint model of Open Babel ignores all charge information and all hydrogens, it generates the same fingerprints for NAD+ and NADH, which makes the total distance between

overall educt state and overall product state very small. That means that the heuristic would prefer reactions that change little as possible, as most changes introduce a higher distance to both overall educt and product state. Histidine, on the other hand, changes its aromaticity upon protonation according to the current model, which increases the distance of the first intermediate state to the overall product state even further. If the aromaticity correction model was fixed, the enzymatic state after the second step of the mechanism would certainly have a much smaller distance to the overall product state than does the overall educt state because in the second step NADH is generated.

Of course, one could argue that the addition of cysteine in the first step activates the part of the molecule that should be modified anyway. But using such observations for a heuristic would mean to guess an atom map before even starting the program. There are two arguments against such a procedure: First of all, relying on a guessed atom map would direct the search towards paths that correspond to this guessed atom map, which could create a bias in the results. After all, one of the goals of MechSearch is to find an atom map that does not correspond to the minimal chemical distance but to a plausible reaction mechanism. Secondly, calculating the optimal atom map as an initial guess can be computationally very expensive. A better approach would be to simply not include parts of the substrate for which the atom map is certain into the input or to use class labels in the SMILES input to make them unreactive.

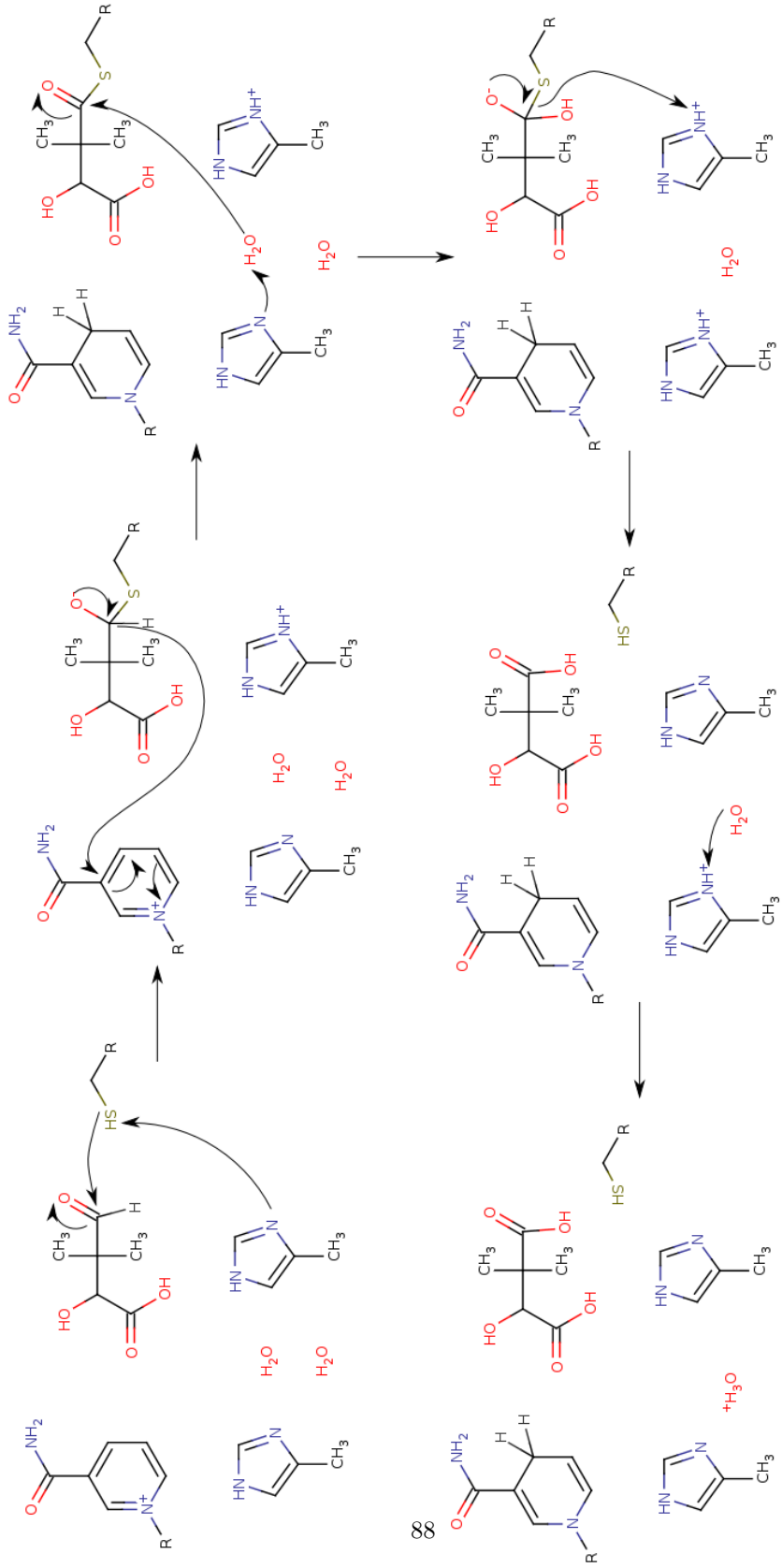


Figure 6.1: Proposed reaction mechanism for the oxidation of the aldehyde to the carboxylic acid. Note that the cofactor is used in the way of a substrate in this reaction and is thus not restored after the reaction.

Part IV

Conclusion and outlook

6.1 Outlook - Possible additions to MechSearch

While the heuristic implemented in this thesis works well, there is still room for improvement.

In order to generate a more accurate heuristic one could think of implementing some sort of reactivity measure in a CAMOE-like fashion[29]. However one would first need to find out how much the active site actually changes the reactivity of the substrate molecules. If the change is substantial in many cases, one would probably need a lot of knowledge about the active site geometry to predict the change in reactivity.

A plausible criterion to determine the correct mechanistic pathway would be the step with the highest activation energy. The pathway with the lowest maximal activation energy would be the best. This corresponds to the observation that one reaction step is usually rate-determining. Unfortunately the activation energy is hard to estimate, especially in the context of enzymatic reactions. A lower bound for the activation energy of a mechanistic step would certainly be the maximum of 0 and the energy difference covered by the reaction step. An upper bound could be the dissociation energy of all bonds that have to be broken in order to go from educts to products. While there have been attempts to estimate the activation energy from the graph representation of the molecules[100], those are only rough estimates and only work for certain types of mechanisms. Thus one way to calculate the activation energy would be to perform QM/MM[101] precalculations in order to associate an activation energy with each rule. Furthermore one would have to predict the modification of these activation energies by the groups attached to the active core and by additional spectator molecules. However, to allow the automatic extension of the knowledge base one would have to define a protocol that allows to calculate these values without the need of manual corrections. Furthermore if classes of substituents and electrostatic spectator molecules are used to modify the activation energy, one would have to make sure that any possible molecule falls into exactly one class and that this classification can be done quickly by the computer.

If, however, only a single, pre-calculated, rough estimate of the activation energy was used for each reaction rule, this could still be used for a heuristic: One could assume that, depending on the overall change in energy and complexity, a minimal number of rules with high activation energy should be found in an optimal reaction path.

Another idea would be to apply machine learning techniques, as has been done for general reaction predictions[32], to the special case of enzymatic reactions.

Right now, all geometric information is ignored. It would, however, be interesting to construct a relative geometry between molecules on the fly during the search for the mechanism. Whenever two molecules react with each other in the active center of an enzyme, one could define a maximal distance between them as a constraint. If different reaction pathways lead to the same state, the constraints of at least one of those pathways would have to be fulfilled. After a few iteration steps contradicting constraints might be found, as we only allow three dimensions for the positioning of molecules. Contradicting constraints could be a good criterion to rule out some reactions. Together with the final pathway one could then construct an approximate geometry for the active center from the constraints.

A problem that occurred several times was aromaticity correction. Sometimes

aromaticity correction of rule subgraph patterns and corresponding molecules yielded different results. Once a fast implementation of the algorithm to derive unique SMILES for mesomeric molecules[89] is available inside the GGL, it would be good to separate aromaticity perception and SMILES generation.

Furthermore it will be very interesting to explore different parts of the chemical space with MechSearch, once the GGL is extended to support valence changes (e.g. of sulfur), carbo cations and complex bonds.

6.2 Conclusion and possible applications

For this thesis, MechSearch, a computer program to calculate reaction mechanisms as synthesis planning problem by bidirectional breadth first search was developed. This program uses the Graph Grammar Library[57] to model molecules as graphs and chemical reactions as graph rewrite rules. By assuming that only a small number of instances of each molecule can coexist at the same time in the finite space of the enzyme's active center, a dramatic increase in search speed was achieved compared to the less specific tool toyChem that previously existed.

As reaction rules for the algebraic chemistry used, elementary reactions were exacted from the MACiE database[85]. An atom map was calculated for these reactions and the extended reaction core that contains the necessary functional groups was calculated. The rules generated this way form a knowledge base for the program MechSearch. In total 884 rules were successfully generated. A clustering approach was used to remove rules that could be expressed by more general rules. After clustering, 711 rules were left, 135 of which were hand curated.

Heuristics were developed for MechSearch to further increase the search speed.

The first heuristic is based on counting molecules that are present at the current state of the search as well as in the initial or final state of the reaction, respectively. This heuristic only increases the search speed a little, but it yields practically no false negatives. It should thus always be used.

The second heuristic calculates a distance between the current enzymatic state and the desired state of the products. This distance used the Tanimoto coefficient between fingerprints as underlying metric, but works for multisets of molecules. With this heuristic the user can trade search time for higher risk of false negatives. This heuristic was evaluated with two different knowledge bases and with additional molecules as disturbance.

A third heuristic criterion that relies on energy values from the Jankowski group contribution method[78] was outlined and can easily be implemented once the energy calculation method has tabulated values for a wider variety of fragments.

Unfortunately, despite all efforts, no certain criterion could be found that is minimized (or maximized) by the correct reaction path. Thus no classical bounded breadth first search could be implemented.

Some reactions were exhaustively calculated. The results show that beside the correct reaction path, several similar paths exist that differ from the correct path in a single proton transfer. These different paths could simply be regarded as wrong artifacts of our model of algebraic chemistry. They could, however, also inspire us to think differently about chemical reactions and reaction mechanisms: On the molecular level everything is moving all the time, different states are in constant equilibrium. Thinking of a reaction mechanism not as a single path from one molecule to another but as a highly connected network with many similar paths and many equilibria thus seems very fitting. The question arises which are the key steps that cannot be circumvented. To address this question, a program was written that finds a representative short path for each reaction rule.

Finally, for some reactions of orphan enzymes found in ORENZA[86] a reaction mechanism was proposed by the program MechSearch. These mechanisms are all chemically plausible. For each elementary reaction in these proposed mechanisms,

the program MechSearch also tells the user the number of the corresponding reactions in MACiE.

Thus MechSearch can be used mainly for two tasks: First, as a first step in elucidation of an enzymatic mechanism. Even though scientists who work in the field of enzyme mechanisms would be able to propose a mechanism similar to the one proposed by MechSearch on a piece of paper if they take enough time, MechSearch could still be handy, as it not only proposes a reaction mechanism, but also gives the user the database ids of all used elementary reactions. Furthermore, given a large enough knowledge base, the program could propose mechanisms with rare reactions a chemist would maybe not think of in the first place.

The second task MechSearch can be used for is probably more important, as for this task MechSearch could not be replaced by a piece of paper and a scientist, but only by dozens of students and tons of paper, namely for batch applications. For reconstruction and analysis of metabolic networks one needs lots of atom mapped reactions. As each proposed reaction mechanism corresponds to a certain atom map, MechSearch can be used to get an atom map that specifically fits enzymatic reactions. Furthermore it can be used to screen whether reactions between molecules that currently are not linked in an incomplete metabolic network are plausible.

In an artificial chemistry MechSearch could be modified and supplied with a small knowledge base to model an enzyme and generate all possible products of given substrates.

Bibliography

[1] Michael F. Lynch and Peter Willett. "George Vladutz, 1928-1990". In: *Journal of Chemical Information and Computer Sciences* 30.4 (1990), pp. 349–349. DOI: 10.1021/ci00068a001. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci00068a001. URL: http://pubs.acs.org/doi/abs/10.1021/ci00068a001.

[2] Elias J. Corey. "General Methods for the Construction of Complex Molecules". In: *Pure Appl Chem* 14 (1967), pp. 19–37.

[3] Elias J. Corey. "The Logic of Chemical Synthesis: Multistep Synthesis of Complex Carbogenic Molecules (Nobel Lecture)". In: *Angew Chem Int Ed* 30.5 (1991), pp. 455–612.

[4] Matthew H. Todd. "Computer-aided organic synthesis". In: *Chemical Society Reviews* 34 (2005), pp. 247–266.

[5] Anthony Cook. "Computer-aided synthesis design: 40 years on". In: *Advanced Reviews* (2011).

[6] Grazyna Nowak and Grzegorz Fic. "Generation of Chemical Transformations: Reaction Pathways Prediction and Synthesis Design". In: *Statistical Modelling of Molecular Descriptors in QSAR/QSPR*. Chap. 15, pp. 393–425.

[7] PENSAK DAVID A. and COREY E. J. "LHASA—Logic and Heuristics Applied to Synthetic Analysis". In: *Computer-Assisted Organic Synthesis*. Chap. 2, pp. 1–32. DOI: 10.1021/bk-1977-0061.ch001. eprint: http://pubs.acs.org/doi/pdf/10.1021/bk-1977-0061.ch001. URL: http://pubs.acs.org/doi/abs/10.1021/bk-1977-0061.ch001.

[8] E. J. Corey et al. "Techniques for perception by a computer of synthetically significant structural features in complex molecules". In: *Journal of the American Chemical Society* 94.2 (1972), pp. 431–439. DOI: 10.1021/ja00757a021. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja00757a021. URL: http://pubs.acs.org/doi/abs/10.1021/ja00757a021.

[9] E. J. Corey, Richard D. Cramer, and W. Jeffrey Howe. "Computer-assisted synthetic analysis for complex molecules. Methods and procedures for machine generation of synthetic intermediates". In: *Journal of the American Chemical Society* 94.2 (1972), pp. 440–459. DOI: 10.1021/ja00757a022. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja00757a022. URL: http://pubs.acs.org/doi/abs/10.1021/ja00757a022.

[10] *Retrosynthetic strategies implemented in LHASA*. Radboud Universiteit Nijmegen. URL: http://cheminf.cmbi.ru.nl/cheminf/olp/strat.shtml.

[11] E. J. Corey et al. "General methods of synthetic analysis. Strategic bond disconnections for bridged polycyclic structures". In: *Journal of the American Chemical Society* 97.21 (1975), pp. 6116–6124. DOI: 10.1021/ja00854a026. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja00854a026. URL: http://pubs.acs.org/doi/abs/10.1021/ja00854a026.

[12] Peter Johnson. "Starting Material orientated Retrosynthetic Analysis in the LHASA Program. 1. General Description". In: *J. Chem. Inf. Comput. Sci.* (1992).

[13] E. J. Corey et al. "Computer-assisted synthetic analysis. Selection of protective groups for multistep organic syntheses." In: *The Journal of Organic Chemistry* 50.11 (1985), pp. 1920–1927. DOI: 10.1021/jo00211a027. eprint: http://pubs.acs.org/doi/pdf/10.1021/jo00211a027. URL: http://pubs.acs.org/doi/abs/10.1021/jo00211a027.

[14] Daren Krebsbach, Herbert Gelernter, and Scott McN. Sieburth. "Distributed Heuristic Synthesis Search". In: *Journal of Chemical Information and Computer Sciences* 38.4 (1998), pp. 595–604. DOI: 10.1021/ci970115v. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci970115v. URL: http://pubs.acs.org/doi/abs/10.1021/ci970115v.

[15] Herbert Gelernter, J. Royce Rose, and Chyouhwa Chen. "Building and refining a knowledge base for synthetic organic chemistry via the methodology of inductive and deductive machine learning". In: *Journal of Chemical Information and Computer Sciences* 30.4 (1990), pp. 492–504. DOI: 10.1021/ci00068a023. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci00068a023. URL: http://pubs.acs.org/doi/abs/10.1021/ci00068a023.

[16] James and Law. "Route Designer: A Retrosynthetic Analysis Tool Utilizing Automated etrosynthetic Rule Generation". In: *J. Chem. Inf. Model.* 49 (2009), pp. 593–602.

[17] Ivar Ugi et al. "Computer-Assisted Solution of Chemical Problems—The Historical Development and the Present State of the Art of a New Discipline of Chemistry". In: *Angewandte Chemie International Edition in English* 32.2 (1993), pp. 201–227. ISSN: 1521-3773. DOI: 10.1002/anie.199302011. URL: http://dx.doi.org/10.1002/anie.199302011.

[18] Robert Höllering et al. "Simulation of Organic Reactions: From the Degradation of Chemicals to Combinatorial Synthesis". In: *Journal of Chemical Information and Computer Sciences* 40.2 (2000). PMID: 10761155, pp. 482–494. DOI: 10.1021/ci990433p. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci990433p. URL: http://pubs.acs.org/doi/abs/10.1021/ci990433p.

[19] J. Gasteiger et al. *Elaboration of Reactions for Organic Synthesis. EROS A Program for Reaction Prediction*. URL: http://www2.ccc.uni-erlangen.de/software/eros/index.html.

[20] James B. Hendrickson, David L. Grier, and A. Glenn Toczko. "A logic-based program for synthesis design". In: *Journal of the American Chemical Society* 107.18 (1985), pp. 5228–5238. DOI: 10.1021/ja00304a033. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja00304a033. URL: http://pubs.acs.org/doi/abs/10.1021/ja00304a033.

[21] James B. Hendrickson. "Organic Synthesis in the Age of Computers". In: *Angewandte Chemie International Edition in English* 29.11 (1990), pp. 1286–1295. ISSN: 1521-3773. DOI: 10.1002/anie.199012861. URL: http://dx.doi.org/10.1002/anie.199012861.

[22] Clemens Jochum, Johann Gasteiger, and Ivar Ugi. "The Principle of Minimum Chemical Distance (PMCD)". In: *Angewandte Chemie International Edition in English* 19.7 (1980), pp. 495–505. ISSN: 1521-3773. DOI: 10.1002/anie.198004953. URL: http://dx.doi.org/10.1002/anie.198004953.

[23] Wolfgang Schubert and Ivar Ugi. "Constitutional symmetry and unique descriptors of molecules". In: *Journal of the American Chemical Society* 100.1 (1978), pp. 37–41. DOI: 10.1021/ja00469a006. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja00469a006. URL: http://pubs.acs.org/doi/abs/10.1021/ja00469a006.

[24] Ivar Ugi and Alf Dengler. "The algebraic and graph theoretical completion of truncated reaction equations". English. In: *Journal of Mathematical Chemistry* 9.1 (1992), pp. 1–10. ISSN: 0259-9791. DOI: 10.1007/BF01172925. URL: http://dx.doi.org/10.1007/BF01172925.

[25] J. Gasteiger and T Kleinöder. *Parameter Estimation for the Treatement of Reactivity Applications - The PTRA package*. URL: http://www2.ccc.uni-erlangen.de/software/petra/index.html.

[26] Lingran Chen and Johann Gasteiger. "Knowledge Discovery in Reaction Databases: Landscaping Organic Reactions by a Self-Organizing Neural Network". In: *Journal of the American Chemical Society* 119.17 (1997), pp. 4033–4042. DOI: 10.1021/ja960027b. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja960027b. URL: http://pubs.acs.org/doi/abs/10.1021/ja960027b.

[27] H. Maarten Vinkers et al. "SYNOPSIS: SYNthesize and OPtimize System in Silico". In: *Journal of Medicinal Chemistry* 46.13 (2003). PMID: 12801239, pp. 2765–2773. DOI: 10.1021/jm030809x. eprint: http://pubs.acs.org/doi/pdf/10.1021/jm030809x. URL: http://pubs.acs.org/doi/abs/10.1021/jm030809x.

[28] Liliana Félix. "Efficient Reconstruction of Metabolic athways by Bidirectional Chemical Search". In: *Bull of Mat Biol* 71 (2009), pp. 750–769.

[29] W. L. Jorgensen et al. "CAMEO: a program for the logical prediction of the products of organic reactions". In: *Pure Appl. Chem.* 62 (10 1990), pp. 1921–1932. URL: http://dx.doi.org/10.1351/pac199062101921.

[30] Jonathan H. Chen. "No electron left behind: A Rule-based Expert System To Predict Chemical Reactions and Reaction Mechanism". In: *J Chem Inf Model* 49 (2009), pp. 2034–2043.

[31] *Daylight Theory Manual*. Daylight Version 4.9. Daylight Chemical Information Systems, Inc. 2011. eprint: www.daylight.com/dayhtml/doc/theory/index.pdf. URL: http://www.daylight.com/dayhtml/doc/theory/index.html.

[32] Matthew A. Kayala et al. "Learning to Predict Chemical Reactions". In: *Journal of Chemical Information and Modeling* 51.9 (2011), pp. 2209–2222. DOI: 10.1021/ci200207y. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci200207y. URL: http://pubs.acs.org/doi/abs/10.1021/ci200207y.

[33] Adalbert Kerber et al. "Molecules in Silico: A Graph Description of Chemical Reactions". In: *Journal of Chemical Information and Modeling* 47.3 (2007). PMID: 17532665, pp. 805–817. DOI: 10.1021/ci600470q. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci600470q. URL: http://pubs.acs.org/doi/abs/10.1021/ci600470q.

[34] *Open Smiles Project*. URL: www.opensmiles.org.

[35] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of Chemical Information and Computer Sciences* 28.1 (1988), pp. 31–36. DOI: 10.1021/ci00057a005. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci00057a005. URL: http://pubs.acs.org/doi/abs/10.1021/ci00057a005.

[36] *Open Babel: The Open Source Chemistry Toolbox*. URL: www.openbabel.org.

[37] Rajarshi Guha et al. "The Blue Obelisk - Interoperability in Chemical Informatics". In: *Journal of Chemical Information and Modeling* 46.3 (2006), pp. 991–998. DOI: 10.1021/ci050400b. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci050400b. URL: http://pubs.acs.org/doi/abs/10.1021/ci050400b.

[38] David Weininger, Arthur Weininger, and Joseph L. Weininger. "SMILES. 2. Algorithm for generation of unique SMILES notation". In: *Journal of Chemical Information and Computer Sciences* 29.2 (1989), pp. 97–101. DOI: 10.1021/ci00062a008. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci00062a008. URL: http://pubs.acs.org/doi/abs/10.1021/ci00062a008.

[39] Shinsaku Fujita. "Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts". In: *Journal of Chemical Information and Computer Sciences* 26.4 (1986), pp. 205–212. DOI: 10.1021/ci00052a009. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci00052a009. URL: http://pubs.acs.org/doi/abs/10.1021/ci00052a009.

[40] Shinsaku Fujita. "Description of organic reactions based on imaginary transition structures. 2. Classification of one-string reactions having an even-membered cyclic reaction graph". In: *Journal of Chemical Information and Computer Sciences* 26.4 (1986), pp. 212–223. DOI: 10.1021/ci00052a010. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci00052a010. URL: http://pubs.acs.org/doi/abs/10.1021/ci00052a010.

[41] Masanori Arita. "The metabolic world of Escherichia coli is not small". In: *Proceedings of the National Academy of Sciences of the United States of America* 101.6 (2004), pp. 1543–1547. DOI: 10.1073/pnas.0306458101. eprint: http://www.pnas.org/content/101/6/1543.full.pdf+html. URL: http://www.pnas.org/content/101/6/1543.abstract.

[42] Masaaki Kotera et al. "Computational Assignment of the EC Numbers for Genomic-Scale Analysis of Enzymatic Reactions". In: *Journal of the American Chemical Society* 126.50 (2004). PMID: 15600352, pp. 16487–16498. DOI: 10.1021/ja0466457. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja0466457. URL: http://pubs.acs.org/doi/abs/10.1021/ja0466457.

[43] Markus Leber et al. "Automatic assignment of reaction operators to enzymatic reactions". In: *Bioinformatics* 25.23 (2009), pp. 3135–3142. DOI: 10.1093/bioinformatics/btp549. eprint: http://bioinformatics.oxfordjournals.org/content/25/23/3135.full.pdf+html. URL: http://bioinformatics.oxfordjournals.org/content/25/23/3135.abstract.

[44] L. Felix, F. Rosselló, and G. Valiente. "Optimal artificial chemistries and metabolic pathways". In: *Computer Science, 2005. ENC 2005. Sixth Mexican International Conference on*. 2005, pp. 298–305. DOI: 10.1109/ENC.2005.30.

[45] Christoph Flamm et al. "Evolution of metabolic networks: a computational frame-work". In: *Jornal of Systems Chemistry* (2010).

[46] *CTfile Formats*. December 2011. Accelrys Scientific and Technical Support. 2011. eprint: http://download.accelrys.com/freeware/ctfile-formats/ctfile-formats.zip. URL: http://download.accelrys.com/freeware/ctfile-formats/.

[47] Eric L. First, Chrysanthos E. Gounaris, and Christodoulos A. Floudas. "Stereochemically Consistent Reaction Mapping and Identification of Multiple Reaction Mechanisms through Integer Linear Optimization". In: *Journal of Chemical Information and Modeling* 52.1 (2012), pp. 84–92. DOI: 10.1021/ci200351b. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci200351b. URL: http://pubs.acs.org/doi/abs/10.1021/ci200351b.

[48] John W. Raymond, Eleanor J. Gardiner, and Peter Willett. "RASCAL: Calculation of graph similarity using maximum common edge subgraphs". In: *The Computer Journal* 45 (2002), p. 2002.

[49] Willett P Raymond JW. "Maximum common subgraph isomorphism algorithms for the matching of chemical structures". In: *J Comput Aided Mol Des* (2002). DOI: 10.1023/a:1021271615909.

[50] L. Félix and G. Valiente. "Efficient Validation of Metabolic Pathway Databases". In: *Proc. 6th Int. Symp. Computational Biology and Genome Informatics* (2005), pp. 1209–1212. URL: http://www.lsi.upc.edu/~valiente/abs-cbgi-2005.pdf.

[51] Tatsuya Akutsu. "Efficient Extraction of Mapping Rules of Atoms from Enzymatic Reaction Data". In: *Journal of Computational Biology* 11 (2-3 2004), pp. 449–462. DOI: 10.1089/1066527041410337.

[52] Torsten Blum and Oliver Kohlbacher. "Using Atom Mapping Rules for an Improved Detection of Relevant Routes in Weighted Metabolic Networks". In: *Journal of Computational Biology* 15 (6 2008), pp. 565–576. DOI: doi:10.1089/cmb.2008.0044.

[53] Heinz Ekker. "Automatic Extraction of Graph Rewrite Rules from Biochemical Reactions". Diploma Thesis. FH Campus Wien - Bioengeneering, 2010.

[54] Robert Körner and Joannis Apostolakis. "Automatic Determination of Reaction Mappings and Reaction Center Information. 1. The Imaginary Transition State Energy Approach". In: *Journal of Chemical Information and Modeling* 48.6 (2008). PMID: 18533713, pp. 1181–1189. DOI: 10.1021/ci7004324. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci7004324. URL: http://pubs.acs.org/doi/abs/10.1021/ci7004324.

[55] Mike Koop and Hardy Moock. *Lineare Optimierung - Eine anwendungsorientierte Einführung in Operations Research*. Springer-Verlag, Berlin Heidelberg, 2008. ISBN: 978-3-8274-1897-5.

[56] Christoph Flamm and Daniel Merkle. *private communication*. 2013.

[57] Martin Mann, Heinz Ekker, and Christoph Flamm. "The Graph Grammar Library - a generic framework for chemical graph rewrite systems". In: *Theory and Practice of Model Transformations, Proc. of ICMT 2013*. Ed. by Keith Duddy and Gerti Kappel. Vol. 7909. LNCS. Extended abstract and poster at ICMT, full article at arXiv. Budapest, HU: Springer, 2013, pp. 52–53. ISBN: 978-3-642-38882-8. DOI: 10.1007/978-3-642-38883-5_5.

[58] Felix H. Reisen, Gisbert Schneider, and Ewgenij Proschak. "Reaction-MQL: Line Notation for Functional Transformation". In: *Journal of Chemical Information and Modeling* 49.1 (2009), pp. 6–12. DOI: 10.1021/ci800215t. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci800215t. URL: http://pubs.acs.org/doi/abs/10.1021/ci800215t.

[59] Jeremy Siek, Lie-Quan Lee, and Andrew Lumsdaine. *The Boost Graph Library*. Indiana University. URL: http://www.boost.org/doc/libs/1_54_0/libs/graph/doc/index.html.

[60] Michael Himsolt. *GML: A portable Graph File Format [online]*. Universität Passau. URL: http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-technical-report.pdf.

[61] Christoph Flamm and Martin Mann. *GGL Tutorial: Graph Rewrite Rules [online]*. Institute for Theoretical Chemistry, Univarsity of Vienna and Bioinformatics group, University of Freiburg. 2013. URL: http://www.tbi.univie.ac.at/software/GGL/Tutorials/tutorial-rules.pdf.

[62] Peter Willett, John M. Barnard, and Geoffrey M. Downs. "Chemical Similarity Searching". In: *Journal of Chemical Information and Computer Sciences* 38.6 (1998), pp. 983–996. DOI: 10.1021/ci9800211. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci9800211. URL: http://pubs.acs.org/doi/abs/10.1021/ci9800211.

[63] Johann Gasteiger et al. “Similarity concepts for the planning of organic reactions and syntheses”. In: *Journal of Chemical Information and Computer Sciences* 32.6 (1992), pp. 700–712. DOI: 10.1021/ci00010a018. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci00010a018. URL: http://pubs.acs.org/doi/abs/10.1021/ci00010a018.

[64] Hina Patel et al. “Knowledge-Based Approach to de Novo Design Using Reaction Vectors”. In: *Journal of Chemical Information and Modeling* 49.5 (2009). PMID: 19382767, pp. 1163–1184. DOI: 10.1021/ci800413m. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci800413m. URL: http://pubs.acs.org/doi/abs/10.1021/ci800413m.

[65] H. Späth. “Partitionierende Cluster-Analyse bei Binärdaten am Beispiel von bundesdeutschen Hochschulen und Diplomstudiengängen”. German. In: *Zeitschrift für Operations Research* 21.4 (1977), B85–B96. ISSN: 0340-9422. DOI: 10.1007/BF01918172. URL: http://dx.doi.org/10.1007/BF01918172.

[66] AlanH. Lipkus. “A proof of the triangle inequality for the Tanimoto distance”. English. In: *Journal of Mathematical Chemistry* 26.1-3 (1999), pp. 263–265. ISSN: 0259-9791. DOI: 10.1023/A:1019154432472. URL: http://dx.doi.org/10.1023/A%3A1019154432472.

[67] M. Deza and E. Deza. *Encyclopedia of Distances*. Encyclopedia of Distances. Springer-Verlag Berlin Heidelberg, 2009. ISBN: 9783642002342. URL: http://books.google.at/books?id=LXEezzccwcoC.

[68] S. M. Turner. “Multiset metrics on bounded spaces”. In: *ArXiv e-prints* (Sept. 2011). arXiv:1109.4930 [math.MG].

[69] Steven H. Bertz. “Complexity of synthetic routes: Linear, convergent and reflexive syntheses”. In: *New J. Chem.* 27 (5 2003), pp. 870–879. DOI: 10.1039/B210844P. URL: http://dx.doi.org/10.1039/B210844P.

[70] Steven H. Bertz. “The first general index of molecular complexity”. In: *Journal of the American Chemical Society* 103.12 (1981), pp. 3599–3601. DOI: 10.1021/ja00402a071. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja00402a071. URL: http://pubs.acs.org/doi/abs/10.1021/ja00402a071.

[71] James B. Hendrickson, Ping Huang, and A. Glenn Toczko. “Molecular complexity: a simplified formula adapted to individual atoms”. In: *Journal of Chemical Information and Computer Sciences* 27.2 (1987), pp. 63–67. DOI: 10.1021/ci00054a004. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci00054a004. URL: http://pubs.acs.org/doi/abs/10.1021/ci00054a004.

[72] Steven H. Bertz, Steven H. Bertz, and Toby J. Sommer. “Rigorous mathematical approaches to strategic bonds and synthetic analysis based on conceptually simple new complexity indices”. In: *Chem. Commun.* (24 1997), pp. 2409–2410. DOI: 10.1039/A706192G. URL: http://dx.doi.org/10.1039/A706192G.

[73] H. W. Whitlock. "On the Structure of Total Synthesis of Complex Natural Products". In: *The Journal of Organic Chemistry* 63.22 (1998), pp. 7982–7989. DOI: 10.1021/jo9814546. eprint: http://pubs.acs.org/doi/pdf/10.1021/jo9814546. URL: http://pubs.acs.org/doi/abs/10.1021/jo9814546.

[74] René Barone and Michel Chanon. "A New and Simple Approach to Chemical Complexity. Application to the Synthesis of Natural Products". In: *Journal of Chemical Information and Computer Sciences* 41.2 (2001), pp. 269–272. DOI: 10.1021/ci000145p. eprint: http://pubs.acs.org/doi/pdf/10.1021/ci000145p. URL: http://pubs.acs.org/doi/abs/10.1021/ci000145p.

[75] Krisztina Boda and A. Peter Johnson. "Molecular Complexity Analysis of de Novo Designed Ligands". In: *Journal of Medicinal Chemistry* 49.20 (2006), pp. 5869–5879. DOI: 10.1021/jm050054p. eprint: http://pubs.acs.org/doi/pdf/10.1021/jm050054p. URL: http://pubs.acs.org/doi/abs/10.1021/jm050054p.

[76] N. Cohen and S. W. Benson. "Estimation of heats of formation of organic compounds by additivity methods". In: *Chemical Reviews* 93.7 (1993), pp. 2419–2438. DOI: 10.1021/cr00023a005. eprint: http://pubs.acs.org/doi/pdf/10.1021/cr00023a005. URL: http://pubs.acs.org/doi/abs/10.1021/cr00023a005.

[77] M L Mavrovouniotis. "Estimation of standard Gibbs energy changes of biotransformations." In: *Journal of Biological Chemistry* 266.22 (1991), pp. 14440–14445. eprint: http://www.jbc.org/content/266/22/14440.full.pdf+html. URL: http://www.jbc.org/content/266/22/14440.abstract.

[78] Matthew D. Jankowski. "Group Contribution Method for Thermodynamic Analysis of Complex Metabolic Networks". In: *Biophysical Journal* 95 (2008), pp. 1487–1499.

[79] Gordon Hammes. "Flexibility, Diversity, and Cooperativity: Pillars of Enzyme Catalysis". In: *Biochemistry* (2011).

[80] D. E. Koshland. "Application of a Theory of Enzyme Specificity to Protein Synthesis". In: *Proceedings of the National Academy of Sciences* 44.2 (1958), pp. 98–104. eprint: http://www.pnas.org/content/44/2/98.full.pdf+html. URL: http://www.pnas.org/content/44/2/98.short.

[81] Daniel E. Koshland. "The Key–Lock Theory and the Induced Fit Theory". In: *Angewandte Chemie International Edition in English* 33.23-24 (1995), pp. 2375–2378. ISSN: 1521-3773. DOI: 10.1002/anie.199423751. URL: http://dx.doi.org/10.1002/anie.199423751.

[82] Mike Williamson. *How proteins work*. Garland Science, New York, 2012. ISBN: 978-0-8153-4446-9.

[83] E F Pai and G E Schulz. "The catalytic mechanism of glutathione reductase as derived from x-ray diffraction analyses of reaction intermediates." In: *Journal of Biological Chemistry* 258.3 (1983), pp. 1752–7. eprint: http://www.jbc.org/content/258/3/1752.full.pdf+html. URL: http://www.jbc.org/content/258/3/1752.abstract.

[84] *The MACiE database online*. URL: http://www.ebi.ac.uk/thornton-srv/databases/MACiE/.

[85] Gemma L. Holliday et al. "MACiE: exploring the diversity of biochemical reactions". In: *Nucleic Acids Research* (2011). DOI: 10.1093/nar/gkr799. eprint: http://nar.oxfordjournals.org/content/early/2011/11/03/nar.gkr799.full.pdf+html. URL: http://nar.oxfordjournals.org/content/early/2011/11/03/nar.gkr799.abstract.

[86] Olivier Lespinet and Bernard Labedan. "ORENZA: a web resource for studying ORphan ENZyme activities". In: *BMC Bioinformatics* 7.1 (2006), p. 436. ISSN: 1471-2105. DOI: 10.1186/1471-2105-7-436. URL: http://www.biomedcentral.com/1471-2105/7/436.

[87] *ILOG CPLEX Optimizer*. IBM Corp. URL: http://www-03.ibm.com/software/products/us/en/ibmilogcpleoptistud/.

[88] *lpsolve - An open source Mixed Integer Linear Programming (MILP) solver*. URL: http://sourceforge.net/projects/lpsolve/.

[89] Martin Mann and Bernhard Thiel. "Kekule structure enumeration yields unique SMILES." In: *Proceedings of WCB13 - Workshop on Constraint Based Methods for Bioinformatics*. Ed. by Alessandro Dal Palu and Dovier Agostino. Uppsala, Sweden, 2013, pp. 57–65. URL: http://cp2013.a4cp.org/sites/default/files/uploads/WCB13_proceedings.pdf.

[90] Martin Mann. *private communication*. 2013.

[91] Bjarne Stroustrup. *A Tour of C++*. Addison-Wesley, 2013.

[92] Nicolai M. Josuttis. *The C++ Standard Library - A Tutorial and Reference*. 2nd Edition. Addison Wesley Longman, 2012. ISBN: 978-0-321-62321-8.

[93] E.W. Dijkstra. "A note on two problems in connexion with graphs". English. In: *Numerische Mathematik* 1.1 (1959), pp. 269–271. ISSN: 0029-599X. DOI: 10.1007/BF01386390. URL: http://dx.doi.org/10.1007/BF01386390.

[94] *Open Babel: The Open Source Chemistry Toolbox - Fingerprint FP2*. URL: http://openbabel.org/wiki/FP2.

[95] Zhigang Wang et al. "Metallo-β-lactamase: structure and mechanism". In: *Current Opinion in Chemical Biology* 3.5 (1999), pp. 614 –622. ISSN: 1367-5931. DOI: http://dx.doi.org/10.1016/S1367-5931(99)00017-4. URL: http://www.sciencedirect.com/science/article/pii/S1367593199000174.

[96] Olabode I. Asubiojo and John I. Brauman. "Gas phase nucleophilic displacement reactions of negative ions with carbonyl compounds". In: *Journal of the American Chemical Society* 101.14 (1979), pp. 3715–3724. DOI: 10.1021/ja00508a002. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja00508a002. URL: http://pubs.acs.org/doi/abs/10.1021/ja00508a002.

[97] Scott J. Weiner, U. Chandra Singh, and Peter A. Kollman. "Simulation of formamide hydrolysis by hydroxide ion in the gas phase and in aqueous solution". In: *Journal of the American Chemical Society* 107.8 (1985), pp. 2219–2229. DOI: 10.1021/ja00294a003. eprint: http://pubs.acs.org/doi/pdf/10.1021/ja00294a003. URL: http://pubs.acs.org/doi/abs/10.1021/ja00294a003.

[98] Kyoungrim Byun and Jiali Gao. "A combined QM/MM study of the nucleophilic addition reaction of methanethiolate and N-methylacetamide". In: *Journal of Molecular Graphics and Modelling* 18.1 (2000), pp. 50 –55. ISSN: 1093-3263. DOI: http://dx.doi.org/10.1016/S1093-3263(00)00035-8. URL: http://www.sciencedirect.com/science/article/pii/S1093326300000358.

[99] Minoru Kanehisa et al. "KEGG for integration and interpretation of large-scale molecular data sets". In: *Nucleic Acids Research* 40.D1 (2012), pp. D109–D114. DOI: 10.1093/nar/gkr988. eprint: http://nar.oxfordjournals.org/content/40/D1/D109.full.pdf+html. URL: http://nar.oxfordjournals.org/content/40/D1/D109.abstract.

[100] Joseph O. Hirschfelder. "Semi-Empirical Calculations of Activation Energies". In: *The Journal of Chemical Physics* 9.8 (1941), pp. 645–653. DOI: 10.1063/1.1750966. URL: http://link.aip.org/link/?JCP/9/645/1.

[101] Hans Martin Senn and Walter Thiel. "QM/MM studies of enzymes". In: *Current Opinion in Chemical Biology* 11.2 (2007), pp. 182 –187. ISSN: 1367-5931. DOI: http://dx.doi.org/10.1016/j.cbpa.2007.01.684. URL: http://www.sciencedirect.com/science/article/pii/S136759310700021X.

Curriculum Vitae

Name	Bernhard Thiel
Mail	thiel@tbi.univie.ac.at

Education

2011–2013	Master Studies of Chemistry, University of Vienna
2008–2011	BSc of Chemistry, University of Vienna
1999–2007	Classical secondary school Bundesgymnasium Wien IX - Wasagasse

Professional Experience

Oct 2012–Oct 2013	Master thesis in the field of cheminformatics Theoretical Bioinformatics group (TBI), University of Vienna

Publications

2013	Martin Mann and Bernhard Thiel. "Kekule structure enumeration yields unique SMILES" In: *Proceedings of WCB13 - Workshop on Constraint Based Methods for Bioinformatics*. Ed. by Alessandro Dal Palu and Dovier Agostino. Uppsala, Sweden, 2013, pp. 57–65
2013	Jean-Luc Mieusset, Bernhard Thiel, Michael Abraham, Mirjana Pačar, and Udo H. Brinker. "Decomposition of an oxodiazirine: free versus incarcerated within the cavities of two α-cyclodextrins" In: *Tetrahedron Letters* 54.7 (2013), pp. 681 –683

Lebenslauf

Name	Bernhard Thiel
Mail	thiel@tbi.univie.ac.at

Bildungsweg

2011–2013	Masterstudium Chemie, Universität Wien
2008–2011	Bachelorstudium Chemie, Universität Wien
1999–2007	Humanistisches Gymnasium, BG IX Wien - Wasagasse

Arbeitserfahrung

10. 2012 – 10. 2013	Masterarbeit im Gebiet der Chemieinformatik Theoretical Bioinformatics group (TBI), Universität Wien

Publikationen

2013	Martin Mann and Bernhard Thiel. "Kekule structure enumeration yields unique SMILES" In: *Proceedings of WCB13 - Workshop on Constraint Based Methods for Bioinformatics*. Ed. by Alessandro Dal Palu and Dovier Agostino. Uppsala, Sweden, 2013, pp. 57–65
2013	Jean-Luc Mieusset, Bernhard Thiel, Michael Abraham, Mirjana Pačar, and Udo H. Brinker. "Decomposition of an oxodiazirine: free versus incarcerated within the cavities of two α-cyclodextrins" In: *Tetrahedron Letters* 54.7 (2013), pp. 681 –683

Danksagung

Ich möchte mich ganz herzlich bei meinem Betreuer Prof. Dr. Christoph Flamm für die Unterstützung während meiner Masterarbeit bedanken.

Ebenso gilt mein Dank Martin Mann für die hilfreichen Diskussionen über die GGL.

Dank auch an Stefan Hammer, der mir geholfen hat, mich in der Welt von Linux halbwegs zurechtzufinden, und mir gezeigt hat, was für coole Sachen man mit Computern machen kann.

Isabel Heger möchte ich dafür danken, dass es ihr beinahe gelang, mich davon zu überzeugen, dass auch im Englischen die Beistrichsetzung einer gewissen Logik folgt.

Außerdem möchte ich mich bei meinen Eltern, meinem Bruder, meiner ganzen Familie und bei meinen Freunden für die Unterstützung während des Studiums bedanken.

Mein Dank gilt auch meinem 4-jährigen Patenkind Julian, das mich erinnert hat, wie viele wichtigere Dinge es gibt als Computer und wie viel Spaß man am Spielplatz haben kann.

Schließlich gilt mein Dank Prof. Dr. Ivo Hofacker und all meinen Kolleginnen und Kollegen am TBI.