

Consistency analysis and improvement of metabolic databases for the integration of metabolic models



seit 1558

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik

der Friedrich-Schiller-Universität Jena

von Dipl. Bioinf Stephan Richter

geboren am 1983-26-10 in Pößneck

2014 July

1. Reviewer: PD Dr. Peter Dittrich,
Friedrich-Schiller-Universität Jena

2. Reviewer: Dr. Martin Thullner,
Helmholtz-Zentrum für Umweltforschung, UFZ, Leipzig

3. Reviewer: Prof. Dr. Christoph Kaleta,
Christian-Albrechts-Universität, Kiel

Day of the defense: 2015-06-26

Signature from head of PhD committee:

Abstract

Modern approaches of systems biology require the analysis and modeling of large-scale metabolic networks. These models can only be assembled by integrating data from various sources like systems biology markup language (SBML)(32) files and online databases. However, sometimes this integration can prove to be rather challenging, as much information can be hidden in human-readable texts or annotational layers not directly accessible with the help of common methods. Here, it is shown how algebraic analysis can be used to unravel structural information hidden in the kinetic laws of SBML models. Additionally, this work will demonstrate the *Organization Theory* (OT) approach and its application for inconsistency detection on the Biomodels Database (44). The usefulness of combining algebraic analysis and OT is shown by comparing the gathered results with data originating from other methods, like flux balance analysis (FBA)(51). The hidden data show how scientific methods can be prone to an incorrect or incomplete interpretation of the data given as well as their computational format of representation. Complementing the analysis of data given in the flat collection of SBML model files, we are also going to present a tool designed to help identifying microbial communities suited to perform biodegradation tasks. Within the respective section, the preliminaries needed to perform such a task are discussed together with problems that interfere with the automatic solution of large-scale puzzles in the field of metabolic research. Afterwards, problems usually occurring in the work with databases are specified and investigated *in dato*; using the Kyoto Encyclopedia of Genes and Genomes as the main source. After the identification of major classes of inconsistencies, strategies to circumvent these problems by rule-based network descriptions are sketched out. It will become clear that the explicit enumeration of all possible chemical compounds poses huge problems and

might be impracticable for large-scale approaches. A detailed description of the idea of rule-based databases for metabolic and biological data which may ease the annotation of numerous compounds will be given. Subsequently, possible applications are listed, giving examples for reasonably simple models. Furthermore, a new formalism will be presented which might suit the task better than more general formalisms like BGNL, which is indeed a very powerful, yet rather tedious methodology. Finally, we will give an account of the advantages and challenges of networks modeled with the rule-based description introduced in this work.

Zusammenfassung

Moderne Ansätze der Systembiologie basieren auf der Analyse und Modellierung von metabolischen Netzwerken im großen Maßstab. Derartige Modelle sind nur durch das Verweben von Daten aus verschiedenen Quellen erreichbar; beispielsweise Dateien im Systembiologie-Markup-Format (SBML) oder im Internet verfügbare Datenbanken. Die Zusammenführung dieser Daten stellt uns vor große Herausforderungen, da oft wichtige Informationen in für das menschliche Auge gedachten Texten versteckt sind. Teilweise sind Daten auch in Notations-Ebenen versteckt, die sich herkömmlichen Verfahren nicht direkt erschließen. In dieser Schrift wird unter anderem aufgezeigt, wie algebraische Analysen genutzt werden können um strukturelle Informationen freizulegen, die in den Massenwirkungsgesetzen von SBML-Dateien annotiert sind. Des Weiteren wird der Organisationstheorie-Ansatz und dessen Anwendung für die Detektion von Unschlüssigkeiten in der Biomodels-Datenbank demonstriert. Ein Vergleich der Ergebnisse dieser Kombination von algebraischer Analyse und Organisationstheorie mit anderen Methoden wie der Fluss-Balance-Analyse (FBA) soll dann die Nützlichkeit dieses Verfahrens belegen. Die damit zugänglich gemachten versteckten Daten zeigen, wie fehleranfällig wissenschaftliche Methoden sind, wenn die zu Grunde liegenden Daten fehlerbehaftet oder ungeeignet formatiert sind. Die Analyse der Biomodels-Datenbank, einer einfachen Sammlung von Modellen im SBML-Format, wird ergänzt durch ein Programm, das entworfen wurde um bestimmte Bakteriengemeinschaften zu ergründen: Diese Bakteriengemeinschaften sollen genutzt werden um auf biologischem Wege Altlasten zu vermindern. In dem entsprechenden Abschnitt dieser Arbeit werden die dafür notwendigen Voraussetzungen erörtert. Des Weiteren wird auf die Schwierigkeiten eingegangen, die unweigerlich auftreten, wenn versucht wird eine

automatisierte Lösung für dieses Problem zu finden. Dabei werden problematische Datensätze erörtert, die auch bei anderen Ansätzen mit großem Datenvolumen zu Schwierigkeiten führen. Um tiefer in die Welt dieser Probleme einzutauchen wird in einem weiteren Kapitel die Kyoto Enzyklopädie für Gene und Genome nach Inkonsistenzen durchleuchtet. Es werden Strategien beschrieben, wie mit den gefundenen Fehlern umgegangen werden kann, und diskutiert, wie sich die entsprechenden Fehler vermeiden lassen. Dabei spielt der Umstieg auf regelbasierte Beschreibungen chemischer Reaktionen eine wesentliche Rolle, um eine explizite Auflistung aller Sonderfälle bestimmter Reaktionen zu vermeiden. Entsprechend wird ein Formalismus für parametrisierte Moleküle und Regelbasierte Reaktionen vorgestellt und mögliche Anwendungen postuliert.

To my family.

Acknowledgements

First of all, I want to thank my supervisor and head of our research group, Peter Dittrich. He always gave kind support and contribution to novel ideas for both my work and side-projects. Also, I want to thank Gerd Grünert, Jan Huwald, Peter Kreyssig and all the other members of the Bio Systems Analysis Research Group for their constructive criticism and support. Special thanks go to my external supervisors from Leipzig and Stockholm, namely Florian Centler, Matrin Thullner, Ingo Fetzner for all the time they spent helping me to get my work done. I am also grateful to the other members of the Helmholtz Centre for Environmental Research I was in contact with, as well as the Helmholtz Foundation for funding large parts of the research project. I feel obliged to also mention the local faculty staff, especially Erik Braun and Uwe Richter, who tirelessly helped to keep our computer systems running.

In personal matters my special thanks for their understanding and support in all aspects of life go to my friends, especially to Katharina Blechschmidt and Konstanze Olchewski. I also feel grateful to have all the other friends who helped me to have a life outside the university, most of all Sergey Fukov, Kenny, Stefan Ehrlich and Franziska Hennig, and Hannes Zöllner.

Last but not least I want to mention that all of this would have been impossible without the support of my whole family and my dear girlfriend Johanna.

Contents

LIST OF FIGURES	ix
LIST OF TABLES	xiii
GLOSSARY	xv
1 INTRODUCTION	1
2 CHECKING SBML BIOMODELS USING HIDDEN DATA	5
2.1 Motivation	5
2.2 Methods	7
2.2.1 Chemical Organization Theory	7
2.2.2 Analyzing Reaction Networks With Modifiers	8
2.2.2.1 Step 1: Identifying Sets Of Essential Modifiers	9
2.2.2.2 Step 2: Adapting The Reactions	10
2.2.2.3 Example Application	11
2.3 Organizational Structure Of The ERK/Wnt-signaling Pathway	11
2.4 Large-Scale Analysis Of Bio-Models	15
2.4.1 Resolving Network Inconsistencies	17
2.4.2 Comparison With Flux-based Methods	19
2.5 Conclusion	22
3 NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE	25
3.1 Introduction	25
3.2 Preconditions For Multi-species Network Models	26
3.3 Basic Approach Description	27
3.3.1 Search For Processing Bacteria	27

CONTENTS

3.3.2	Filter Bacteria	28
3.3.3	Optimize Degradation And Determination Of Additional	28
3.3.4	Find Secreted Intermediate Substances	28
3.3.5	Iterative Search For Further Degradation Pathways	29
3.3.6	Finding The Optimal Set Of Bacteria	29
3.4	Implementation	30
3.4.1	Client-Server Architecture	30
3.4.2	The Local Database	30
3.4.2.1	Structure Of The Local Database	30
3.4.2.2	Data Collection	34
3.4.3	Graphical User Interface	35
3.4.4	The Server	43
3.4.5	Clients And Optimization Routines	43
3.4.5.1	Search For Processing Organisms	43
3.4.5.2	Calculate Products	44
3.4.5.3	Calculation Of Additional Maximizing The Set Of Products	44
3.4.5.4	Calculate Seeds	45
3.4.5.5	Calculate Flow Distributions For Given Input/Output	46
3.4.5.6	Paths From “Substances To Degrade” To “Substances To Produce”	46
3.4.5.7	Calculate Additional With Evolutionary Algorithm	47
3.4.6	Exemplary Workflow	47
3.5	Calculation Problems	49
4	INCONSISTENCY ANALYSIS ON THE KEGG DATABASE	51
4.1	Overview	51
4.2	Introduction	52
4.3	Methods	53
4.3.1	Structure Of KEGG	53
4.3.2	Local Data Management	53
4.3.3	Algorithm	54
4.3.3.1	Substance Checking	54

4.3.3.2	Reaction Checking	54
4.4	Analysis Of Database Inconsistencies	55
4.4.1	Database Evolution	55
4.4.2	Inconsistent Compound References	56
4.4.3	Balanced Reactions	57
4.4.4	Indistinct Reactions	57
4.4.5	Unbalanced Reactions	57
4.4.5.1	Inappropriate Annotation	58
4.4.5.2	Polymer Reactions Containing Quantifiers Such As n Or m	58
4.4.5.3	Disappearing Protons	60
4.4.5.4	Other Reactions With Mismatching Atom Counts	61
4.4.5.5	Transmutations As A Peculiar Subclass Of Unbalanced Reactions	62
4.4.6	Reviewing Previously Identified Database Inconsistencies	64
4.5	Towards Rule-Based Metabolic Databases	65
4.5.1	Proposal For A Residue Database	65
4.5.1.1	Compound And Residue Set	65
4.5.1.2	Reactions Using General Molecules And Referencing In- stances	67
4.5.2	Introducing Rules	69
4.5.2.1	Compounds	69
4.5.2.2	Reactions	74
4.6	Summary	79
4.6.1	Additional Database Fields	79
4.6.2	Rule-based Species And Reactions	80
4.6.3	Substituent Collection	80
5	RULE-BASED METABOLIC DATABASES	83
5.1	Classes Of Rulebased Molecules	83
5.2	Formalization Of Rule-based Reactions	84
5.2.1	Parametrized Molecules	85
5.2.2	Formalization Of Reactions	87

CONTENTS

5.2.3	Mapping Between Educts And Products	91
5.2.4	Reactivity Check	93
5.3	Annotation Frameworks For Rulebased Reactions	93
5.3.1	InChI	94
5.3.2	SMILES	94
5.3.3	BNGL	94
5.4	Homopolymers - Polymers Of Variable Length	95
5.4.1	Example 1: Interval Chemistry	96
5.4.1.1	Approximative Algorithm	96
5.4.1.2	Graphical Analysis	97
5.4.1.3	Analytic Constant Time Algorithm	99
5.4.2	Example 2: Coupling Of Equal Chains	101
5.4.3	Example 3: Fatty Acid Degradation	104
5.4.3.1	The Reaction System	104
5.4.3.2	Elementary Flux Mode Analysis	106
5.5	Linear Polymers With Several Building Blocks	113
5.5.1	Fixed Ratio Heteropolymers	113
5.5.2	Aperiodic Linear Polymers	113
5.6	Attachments And Charges	116
5.7	Branched And Cross-linked Structures	117
5.8	Discussion	117
6	CONCLUSION AND OUTLOOK	119
7	SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING	123
7.1	Emulating Flux-based Network Analysis Methods	123
7.2	Extending The Steady-state Lemma To Growth States	124
7.3	Detailed Description Of The SBML Processing Algorithm	126
7.3.1	Libraries Used	126
7.3.2	Overview On The Processing Steps	126
7.3.3	Description Of The Steps	126
7.3.3.1	Reading And Testing The SBML Code	126
7.3.3.2	Searching For Defined Meta-ids	127

7.3.3.3	Building Function And Parameter Look-up Tables . . .	127
7.3.3.4	Analysis Of The Structure Of The Kinetic Laws	127
7.3.3.5	Adapting The Reaction Structure And The Kinetic Laws	130
7.4	Abbreviations	131
7.5	Detailed List Of Networks Of The BioModels Database	131
7.5.1	Curated Branch	132
7.5.2	Non-Curated Branch	138
7.5.3	Discussion	140
8	SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES	143
8.1	Same Substances With Different Formulas	143
8.2	Indistinctive Reactions	144
8.3	Unbalanced Reactions	160
8.4	Transmutations	191
9	SUPPLEMENT: SOURCE CODES FOR INTERVAL CHEMISTRY	
	ALGORITHMS	197
9.1	Logarithmic Approximation	197
9.1.1	Initialization	197
9.1.2	Calculation Methods	198
9.1.3	Main Entry Point	200
9.2	Geometrical Analysis	200
9.2.1	Initialization	200
9.2.2	Main Code	200
9.3	Fast Interval Expansion	202
9.3.1	Initialization	202
9.3.2	Calculation Methods	203
9.3.3	Main Entry Point	204
10	CURRICULUM VITAE	205
	REFERENCES	207

CONTENTS

LIST OF FIGURES

2.1	Example network with 7 species and 3 reaction rules, hasse diagram. . .	8
2.2	Simplified representation of the reaction network from <i>BIOMD149</i> (41).	12
2.3	Hasse diagram of elementary organizations of <i>BIOMD149</i> without processing of the kinetic laws.	13
2.4	Hasse diagram of elementary organizations of <i>BIOMD149</i> after the processing of the kinetic laws.	14
2.5	Histogram of the number of reactive organizations in the models of the BioModels Database.	16
2.6	Simplified representation of the reactions of <i>BIOMD143</i>	18
2.7	Reaction network from <i>BIOMD037</i> modeling the sporulation control in <i>Physarum polycephalum</i>	21
3.1	The logical subunits of the interaction toolbox.	31
3.2	Structure of the Local Database.	32
3.3	The species lists form.	36
3.4	The substances lists.	38
3.5	The optimization parameters tab.	39
3.6	The results tab.	40
3.7	The network view - showing the reactions related to D-Glucose in an E.Coli model	41
3.8	The database tab - allowing to add new models	42
4.1	Examples of Classified Unbalanced Reactions from the KEGG Database	59
4.2	Examples of Unbalanced Reactions not Belonging to any of the Aforementioned Classes	61

LIST OF FIGURES

4.3	Examples of transmutational reactions	62
4.4	Reaction Classes and their Major Subsets	63
4.5	Classification of KEGG Reactions, as Retrieved on May 2013.	64
4.6	Application of the Residue Database for Specifying Generic Species . . .	66
4.7	Reshaped Entry for Alcohol Dehydrogenase Reaction	68
4.8	Nucleobase Compounds used for our Rule-based Database Example . .	70
4.9	A Rule-based Specification of a (Generic) Nucleobase	71
4.10	Substituent Entries for the Generic Substituents “Pyrimidin base” and “Purin base”.	71
4.11	Substituents “(deoxy)Nucleoside” and “(deoxy)Nucleotide”.	73
4.12	Anticipated Nucleic Acid Entry.	73
4.13	Rule-based Compounds “DNA” and “RNA” Using our Suggested Fields for Composition and Crosslinking.	74
4.14	Rule-based Reaction “Nucleotide Activation”.	75
4.15	Rule-based Reaction “Nucleic Acid Elongation”.	76
4.16	Specific Reaction “ATP:dTDP Phosphotransferase” Instantiated from Rxxxx1	77
4.17	Relation of the Entries of the Example Presented to the Rule-based Data Base	78
5.1	Generic scheme for nucleotides and common nucleotides	85
5.2	Visualization of the relation of inflows and outflows: first iteration . . .	98
5.3	Visualization of the relation of inflows and outflows: final.	100
5.4	Exemplary reaction network layouts for a system where n identical chains are coupled for $n \in \{2, \dots, 4\}$	102
5.5	Building blocks for symmetric addition networks, where n identical chains of length m are coupled.	103
5.6	Basic structure of linear saturated acyl-CoA thioesters and correspond- ing names for length up to 7	104
5.7	Main molecule classes involved in the fatty acid degradation cycle. . . .	105
5.8	Overall reaction sheme for the biodegradation of fatty acids.	108
5.9	Reduced reaction scheme.	109
5.10	The three elementary modes found by the algorithm.	112

5.11 Schematic reaction path for creation of a DNA with sequence “GAT-TACA”. 115

7.1 Tree corresponding to the rate law of reaction 3 from the example network 127

7.2 The same tree after application of zero values to the (absent) modifiers. 128

7.3 The right subtree solved to zero, the left subtree was replaced by A 's value, say 3 129

7.4 The right subtree is solved to zero, if the two modifiers $M3$ and $M4$ are absent 129

7.5 In (a), we applied a zero concentration to $\{M3\}$, solved the multiplications by zero (b) and obtained the rate law $v_{R3} = 3 \cdot [M4] \cdot [M5]$ after the application of A 's value 3 (c). 135

7.6 Since the supporting set is the empty set, no modifier has to be moved to the educt or product side, hence also the kinetic laws are left unchanged. 136

7.7 We obtain three supporting modifier sets, leading to three reaction variants. Note the division by 3 in the kinetic laws. 137

LIST OF FIGURES

LIST OF TABLES

2.1	Selected results from the large-scale analysis	19
4.1	Number of Extracted Entries of the KEGG Database at different times	55
4.2	Inconsistent Reactions and Corresponding Strategies	65
4.3	Example of a Residue Database Containing Five Residues used in the Alcohol Dehydrogenase Reaction.	67
5.1	Possible acceptors for the different oxidation reactions within the fatty acid degradation	106
7.1	Abbreviations.	131
7.2	Metabolites of the complete network	132
7.3	Metabolites of the complete network	138
8.1	Dissent Formulas in synonym Compound Entries	143
8.2	Indistinctive reactions in KEGG	144
8.3	Unblanced reactions in KEGG	160
8.4	“Transmutations” reactions in KEGG.	191

GLOSSARY

GLOSSARY

- IN** The set of *natural numbers* including zero (0).
- BNGL** The *BioNetGen Language* (19) is a formal language that allows rule-based modeling of biochemical systems.
- CLOSURE** *Closure* is a concept borrowed from organization theory (\rightarrow OT) and denotes the maximum set of substances reachable from an initial set.
- DB** *Database*
- EC-Number** The *Enzyme Commission-Numbers* provide a four-section numerical classification system for enzymes, sorted by the respective catalytic function.
- EMA** *Elementary Mode Analysis* is an analysis concept for metabolic pathways, which aims to uncover structures of metabolic networks.(64)
- EPA** *Extreme Pathway Analysis* is a mathematical concept for analyzing the minimal flows that can be combined to form steady state metabolic networks.(58)
- ETF** *Electron Transferring Flavoprotein*.
- FAD** *Flavin Adenine Dinucleotide*.
- FBA** *Flux Balance Analysis* is a mathematical method for analysing metabolic networks. In a flux balance analysis, one searches for a single optimal flux distribution maximizing a given criterion, like biomass production. (51)
- FIFO** *First In First Out* is a handling order in which tasks are processed in the same order as they are created.
- ILP** *Integer Linear Programming* is a method of optimization based on integer variables with linear constraints.
- InChI** The *International Chemical Identifier* is a format for the linear textual representation of molecules 5.3.1.
- KEGG** The *Kyoto Encyclopedia of Genes and Genomes* is one of the most famous online databases holding genetic, metabolic and structural information of organisms (37, 38). Whenever data from KEGG is mentioned within this document, I refer to the state of the database on 2013/11/01.
- NAD⁺** *Nicotinamide Adenine Dinucleotide*
- NADP⁺** *Nicotinamide Adenine Dinucleotide Phosphate*
- OT** *Chemical Organization Theory* is a formalism that allows to map complex reaction networks to closed sets of self-maintaining molecules. (17)
- SBML** *Systems Biology Markup Language* is an XML based file format developed for the storage of metabolic network data. (32)
- SMILES** The *Simplified Molecular Input Line Entry Specification* is a format for the linear textual representation of molecules 5.3.2.
- TCK** The *treasure camouflaged key* is a special item to flag additional document contents

GLOSSARY

- URL** *Uniform Resource Location* is the address of a web page. identifier for an entity of a model or the real world.
- URN** *Uniform Resource Name* is a unique

1

INTRODUCTION

Throughout the last centuries, the investigation of biochemical systems and metabolic networks in particular has come a long way; starting with the observation of macroscopic animals and plants and, over the years, moving on to the investigation of smaller living entities. After Hooke's findings(30) and the realization that biological tissues are composed of cells, scientists steadily increased the range of vision of their research. Today, we already reached sub-molecular levels, making various organisms and organic molecules objects of detailed investigations. Within the last years, research on cellular networks of reactions and chemicals, enzymes, and their interdependency with genetic factors, in short metabolic research, achieved astonishing insights and results. Over the last decades, an overwhelming amount of knowledge has been collected and categorized, resembling the tiniest parts of a giant jigsaw puzzle which now awaits being pieced together.

With the help of computer power and smart algorithms, today the focus of biological and microbial research has largely shifted from the investigation of single biological entities to more systemic approaches. This means that whereas in the past single enzymes, proteins, molecules, or reactions largely constituted the focus of research, today we are dealing with proteomic, genomic, or metabolomic data. These *omics* sciences are modern high-tech approaches trying to get a holistic image of biological and biochemical systems. To do so, they both utilize and generate enormous amounts of data in very short time spans, using the capabilities of modern computers. In general, these large-scale approaches have become possible due to the following major reasons:

1. INTRODUCTION

1. In accordance with Moore's law (49), computers have reached a super high integration level and became incredibly fast in the last decades.
2. As a consequence of the high integration and miniaturization, storage capacity has reached sizes allowing for the storage of fine detailed descriptions of any aspect of living organisms.
3. Due to the development of the internet, scientists have the possibilities to collectively acquire new insights and share them globally.

The computer power achieved in the last decades facilitated the collection of detailed information for diverse aspects and created both the demand and the base for large-scale online databases for genomes and molecular data. These databases themselves have become the base for novel approaches taking into account systems of the size of a whole cell. Right now, researchers are just starting to overcome this order of magnitude by creating multi-cellular models and integrating data from various sources.

After all, this new scope also gives birth to new challenges: To be able to assemble large models, one has to be aware of the problems and insufficiencies present in our comprehensive knowledge bases. For large-scale models to be thrustworthy, it is first of all crucial to have a solid, well curated basement. Generally speaking, care has to be taken when entering vague data into storage frameworks, since nobody is able to foresee where this respective data will be reused. Yet, it is of vital importance to be alert when using data from databases, as even in a curated state they might still contain errors or inaccuracies. Thus, methods working with data from these sources should be able to deal with these (possible) flaws.

As modern approaches work with large scale data sets, it is indispensable to keep the knowledge bases used in a consistent state; including the possibility of easy access. In this context *consistency* means that databases should have a well documented composition, all entries should be in a well-defined format and should comprise metadata stating its curation status and confidence level. These demands lead to the development of new methods for investigating inconsistencies within databases, their origin and strategies to overcome current issues. After giving an overview on some influential methods applied within the field of life sciences and the data used for these methods, this thesis introduces new approaches for quality assessment.

Consequently, central questions motivating the research presented in this document are:

- Which important methods are frequently applied in systems biology?
- What are the fields of application for these methods? How could standard methods be utilized to solve environmental problems?
- Which are the problems impeding these methods?
- How can these obstacles be overcome? How can the consistency of large-scale models generated from online databases be assured or how can the level of consistency at least be improved?

In Chapter 2 it is shown how algebraic analysis can be used to unravel structural information hidden in the kinetic laws of models captured in systems biology markup language (SBML)(32). To do so, the *Organization Theory* (OT) approach and its application for inconsistency detection on the Biomodels Database (44) will be demonstrated. Moreover, the usefulness of the combination of algebraic analysis and organization theory is shown by comparing the gathered results with data originating from other methods, like flux balance analysis (FBA)(51). While the hidden data mentioned before are not exactly errors, they show how methods can be prone to an incorrect or incomplete interpretation of the data given as well as their computational format of representation. However, only a selection of important methods will be mentioned in this thesis. For more detailed descriptions of the methods used the interested reader is advised to refer to the relevant publications.

Subsequently, a real world application of methods combined from the field of systems biology and computer science is presented: Chapter 3 describes a tool that was designed to help identifying microbial communities suited to perform biodegradation tasks. Such communities might help to dispose liabilities inherited from chemical industry or waste dumps. Constructing network models of such communities requires integration of data from various sources. Hence, the tool described makes use of a collection of databases rather than using a single database presented as the flat collection of SBML model files. Within this section, the preliminaries needed to perform this integration and operation on the integrated data are discussed. This is complemented with problems

1. INTRODUCTION

that interfere with the automatic solution of large-scale puzzles in the field of metabolic research.

The problems related to databases considered in Chapter 3 are specified and investigated *in dato*; using the Kyoto Encyclopedia of Genes and Genomes as the main source in the next chapter (4). After the identification of major classes of inconsistencies, strategies to circumvent these problems by rule-based network descriptions are sketched out. It is clarified that the explicit enumeration of all possible chemical compounds poses huge problems and might be impracticable for large-scale approaches.

As a rule-based approach will have been only roughly outlined up to this point, a more detailed description of the idea of rule-based databases for metabolic and biological data will be given in Section 5. In this chapter, possible applications are listed, giving examples for reasonably simple models. Furthermore, a new formalism will be presented which might suit the task better than more general formalisms like BGNL, which is indeed a very powerful, yet rather tedious methodology. Finally, this chapter will give an account of the advantages and challenges of networks modeled with the rule-based description introduced in this thesis.

2

CHECKING SBML BIOMODELS USING HIDDEN DATA

2.1 Motivation

There is a growing number of software tools performing calculatory tasks on metabolic networks. Programs implementing methods like FBA, EMA (64), EPA (58) or OT rely on stoichiometric data supplied in SBML files. The first three methods do not require any further knowledge of kinetic laws; and even though those kinetic rules pose influential details for calculating chemical organizations, quite often information about them is indeed unavailable for a large number of reactions. For the purpose of OT, such data need to be unveiled, since the consideration of modifiers and kinetic laws is as crucial as reaction stoichiometric data, especially when it comes to decide whether a reaction can take place or not. Hence, an algorithm was implemented that extracts such hidden information and adjusts reaction laws accordingly.

The following section will begin with an overview over organization theory, followed by the description of our algorithm to extract decisive rules for the calculation of chemical organizations. This tool was then applied by Christoph Kaleta to models from the biomodels database (44). In a second step, the OT results with and without extractive processing were compared to each other. The remainder of this chapter has been published as “Using chemical organization theory for model checking” in

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

Bioinformatics (36).

An important property of chemical organizations is that every steady state and growth state¹ of a network corresponds to a chemical organization ((17) and Supplement). These states we call the limit behavior of a model. However, this property is fulfilled only if a reaction network meets a condition formulated by (20): each reaction has a non-zero flux if and only if all of its educts have a positive concentration. Using this property, OT has already been applied to the prediction of growth phenotypes (10) and the outcome of knockout experiments (34), as well as in the design of chemical programs to solve NP-complete problems (48).

In a recent work we used OT to assess the quality of a genome-scale reaction network of *Escherichia coli* by identifying species and reactions that could not be present in the limit behavior of the model during simulation (11). We concluded that these species and reactions hint at missing knowledge as they were mostly part of pathways starting from or ending in dead-end species. Here we want to extend this approach in two directions. First, we present a method for more accurately predicting the limit behavior of a reaction network if information on reactions kinetics is available. If modeled in Systems Biology Markup Language (SBML)(32), the velocity of a reaction depends on the concentration of its educts, products, and modifiers. A modifier is a species whose concentration affects the reaction velocity but whose concentration itself is not changed by this reaction. Some modifiers, as for example catalysts or activators, are required to be present for a non-zero reaction velocity. However, if we want a reaction to fulfill the Feinberg condition, such modifiers need to be added on its educt and product sides. Hence, since information necessary for the analysis using OT can be hidden in the kinetic laws, we present an algorithm for extracting this information. Second, using this approach, we demonstrate how knowledge of the organizational structure of a reaction network and thus of its limit behavior can help to uncover modeling inconsistencies. These inconsistencies are represented by species as well as by reactions that belong to no organization, indicating either missing knowledge, compounds missing from the specified growth media or modeling errors.

¹As growth state we define a situation where some species accumulate. An example is exponential growth in which, for instance, the overall amount of DNA increases given that there is a continuous supply (inflow) of nutrients.

This work is structured as follows. In Section 2.2 we give a short outline of OT and present an algorithm that modifies the stoichiometric structure in a reaction network such that the Feinberg condition is fulfilled. We use this algorithm in Section 2.3 to demonstrate how these modifications affect the organizational structure of a model of the ERK/Wnt-signaling pathway. In Section 2.4 we use our approach to find inconsistencies in a large-scale analysis of the models of the BioModels Database (44) and compare our results with those obtained by other stoichiometric analysis techniques. Finally, we conclude in Section 2.5.

2.2 Methods

2.2.1 Chemical Organization Theory

We define a reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ by a set of molecular species \mathcal{M} and a set of reaction rules \mathcal{R} . A reaction rule $\rho \in \mathcal{R}$ is defined by the stoichiometric coefficients $l_{i,\rho}$ and $r_{i,\rho}$ denoting the left-hand and right-hand sides of a reaction rule, respectively. Given a reaction rule $\rho \in \mathcal{R}$, we denote the set of reactant species and set of product species by $\text{LHS}(\rho) := \{i \in \mathcal{M} | l_{i,\rho} > 0\}$ and $\text{RHS}(\rho) := \{i \in \mathcal{M} | r_{i,\rho} > 0\}$, respectively. With $\mathbf{N} = (n_{i,\rho}) = (r_{i,\rho} - l_{i,\rho})$, we denote the stoichiometric matrix of $\langle \mathcal{M}, \mathcal{R} \rangle$. W.l.o.g. we assume $\mathbf{v}_\rho \geq 0$; hence a reversible reaction has two entries in v .

Given a set $A \subseteq \mathcal{M}$, its set of reaction rules $\mathcal{R}_A = \{\rho \in \mathcal{R} | \text{LHS}(\rho) \subseteq A\}$, and the corresponding stoichiometric matrix \mathbf{N}_A , we say that A is closed if for all reaction rules $\rho \in \mathcal{R}_A$, $\text{RHS}(\rho) \subseteq A$. Thus, we call A closed if there is no reaction with educts from A producing a species not in A . A is self-maintaining if there exists a strictly positive flux vector $\mathbf{v}' \in \mathbb{R}_{>0}^{|\mathcal{R}_A|}$ such that all species in A are produced at a non-negative rate, that is, $\mathbf{N}_A \mathbf{v}' \geq 0$ (17). A set A that is closed and self-maintaining is called an organization (23). First is the reactive organization. An organization is called reactive if each of its species participates in at least one reaction of that organization. Elementary organizations are reactive organization that cannot be generated as union of other reactive organizations (11).

Because organizations may share the same species, the set of organizations together with the set inclusion \subseteq form a partially ordered set that can be visualized in a Hasse diagram, providing a hierarchical view of the network under consideration: Organizations are vertically arranged by size, with small organizations at the bottom. Two

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

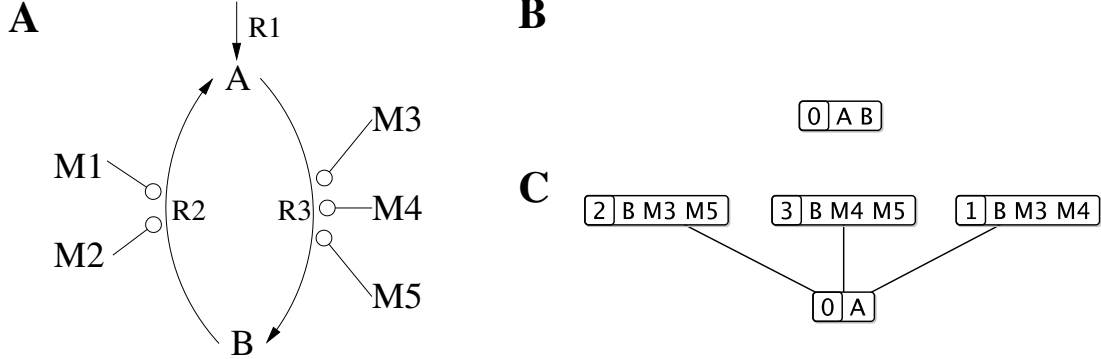


Figure 2.1: **A** Example network (phosphorylation cycle) with 7 species and 3 reaction rules. **B** and **C** Hasse diagrams of elementary organizations of the unprocessed and processed networks respectively. Only species appearing for the first time in each organization are displayed.

organizations are connected by a line if the upper contains the lower organization and no other organization exists between them. For simplicity, only species appearing for the first time, i.e., which are not element of a lower organization, are displayed.

2.2.2 Analyzing Reaction Networks With Modifiers

In this section we introduce an algorithm that allows application of OT to reaction network models containing modifiers. As an example, we use a phosphorylation cycle, a typical motive found in signaling networks (Figure 2.1 **A**). The network consists of 7 molecular species $\mathcal{M} = \{A, B, M1, \dots, M5\}$ and 3 reactions $\mathcal{R} = \{R1, R2, R3\}$.

For the reactions $\mathcal{R} = \{R1 : \emptyset \rightarrow A, R2 : B \rightarrow A, R3 : A \rightarrow B\}$ we assume the following kinetic laws (omitting rate constants and units):

$$\begin{aligned} v_{R1} &= 1 \\ v_{R2} &= [B] (1 + [M1] + [M2]) \\ v_{R3} &= [A] ([M3][M4] + [M3][M5] + [M4][M5]) \end{aligned}$$

This model can be formulated in SBML, with $M1, M2$ being modifiers of reaction $R2$ and $M3, M4, M5$ being modifiers of reaction $R3$, while not appearing as reactants.

Our algorithm consists of 2 steps: First, we examine the kinetic law of each reaction to detect minimal sets of modifiers that are necessary for that reaction to have a positive flux. Then we use this information to adapt a reaction's set of reactants in order to

more faithfully reflect the algebraic structure of the network used for computation of chemical organizations.

2.2.2.1 Step 1: Identifying Sets Of Essential Modifiers

In this first step we identify all minimal supporting modifier sets of each reaction. Given a reaction $\rho \in \mathcal{R}$, a *minimal supporting modifier set* (*supporting set*, for short) is defined as a minimal set of modifiers that need positive concentrations (while all others are absent) to allow reaction ρ to have a positive flux. If at least one of these modifiers is additionally set to a zero concentration, the flux of the reaction is constrained to zero. There might be several possibly overlapping supporting sets. With respect to a certain reaction, a modifier is called *essential* if it is contained in all supporting sets of the reaction.

1. Determination of supporting sets

To decide whether a set of modifiers is a supporting set for a particular reaction, we follow a straightforward approach: If a set of modifiers is a supporting set, a positive concentration of only these modifiers allows a non-zero flux, while a positive concentration of only a proper subset of these modifiers constrains the flux to zero. Following this idea, we implemented FormulaChecker, which tries to compute the velocity of each reaction in terms of modifier concentrations. All variables in the kinetic law that represent undefined parameters or educt or product species are not further resolved; i.e., they are treated as symbols. The modifiers we want to test to determine whether they belong to a supporting set are also treated as symbols. The remaining modifiers are set to zero concentration.

Function calls are resolved by application of their respective parameters, if necessary. Applying FormulaChecker can lead to two different results for the reaction velocity:

(1) The result is zero. In this case the tested modifier set is not a supporting set.

Let $\{M3\}$, for example, be the set to be checked in $R3$. Setting the concentrations of the remaining modifiers to zero results in $v_{R3} = 0$. Thus, $\{M3\}$ is not a supporting set of $R3$. This also applies to the sets $\{M4\}$ and $\{M5\}$.

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

(2) The result is non-zero. Thus, it might be a constant only depending on parameters, or a formula, dependent on variables. Checking $\{M3, M4\}$ in $R3$ yields the kinetic law $v_{R_3} = [A]([M3][M4])$. Since we know that $\{M3\}$, $\{M4\}$, and $\{M5\}$ do not represent supporting sets, $\{M3, M4\}$ has to be a supporting set.

In contrast, if we check the empty set in $R2$ by setting $M1$ and $M2$ to zero values in the kinetic law, we obtain $v_{R_2} = [B]$. In consequence, neither $\{M1\}$ nor $\{M2\}$ represent supporting sets of $R2$; the supporting set is the empty set, and no further tests are required.

2. Finding all supporting sets In order to find all supporting sets of a reaction, the algorithm analyzes the power set of the reaction's set of modifiers to ensure that all supporting sets are found. The sets are checked in increasing size order, trying to avoid testing the whole power set of modifiers: If we find that a set of modifiers is a minimal supporting set, we do not need to test any of its supersets.

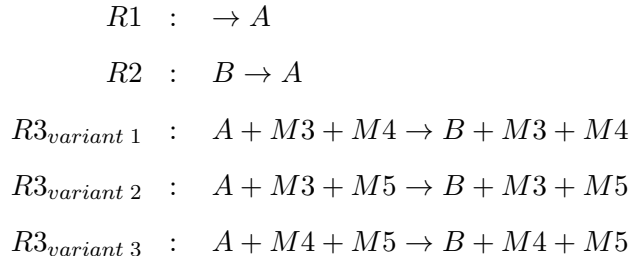
Looking at $R3$ in the example, after the empty set, all single-modifier sets are checked. We find that neither $M3$ nor $M4$ nor $M5$ allow a positive flux if standing alone. In the next step all two-element sets are tested. Since all these sets allow a positive flux of $R3$, but none of the smaller ones, we conclude that $\{M3, M4\}$, $\{M3, M5\}$, and $\{M4, M5\}$ are the supporting sets. In consequence, we do not have to test the superset $\{M3, M4, M5\}$.

2.2.2.2 Step 2: Adapting The Reactions

In the second step, each reaction possessing at least one supporting set is processed. For each supporting set the reaction is duplicated and the modifiers of the supporting set are added as catalysts to the duplicate reaction. Finally, the original reaction is removed from the model. In order to preserve the dynamics of the original model in the processed model, the kinetic law of each of the duplicate reactions is divided by the number of derived reactions, i.e., the number of supporting sets. The duplicate reactions get new names of the form $[old_reaction_name] \text{ variant } [number]$. For our example we obtain the following set of reaction rules $\mathcal{R} = \{R1, R2, R3_{variant\ 1},$

2.3 Organizational Structure Of The ERK/Wnt-signaling Pathway

$R3_{variant\ 2}, R3_{variant\ 3}$ with



For a more detailed outline of the processing of the kinetic laws, see the Supplement.

2.2.2.3 Example Application

Applying the algorithm to our example, we can see several effects of the processing of the kinetic law (see Figures 2.1B and 1C for the Hasse diagrams of elementary organizations). Two trends are superimposed. First, some organizations vanish, including the organization solely containing A and B in the unprocessed network. In the processed network, a reaction still converts B to A . In order to replenish B , one pair of the modifiers $M3$, $M4$, and $M5$ is necessary. Thus, $\{A, B\}$ does not fulfill the self-maintenance condition in the processed network. Second, some organizations appear for the first time, as in the case of the organization containing A in the processed network. In the original network, the set $\{A\}$ was not closed since $R3$ unconditionally produced B from A .

2.3 Organizational Structure Of The ERK/Wnt-signaling Pathway

In order to demonstrate the utility of the incorporation of kinetic laws into the analysis with OT, we analyze the model *BIOMD149*¹ from the BioModels database (44) containing an integrated ERK and Wnt/ β -catenin signaling pathway (Figure 2.2). This model is based on the work of Kim et al. (41), who described a positive feedback loop between these two pathways important in the development of some cancer (41). The positive feedback loop works through a yet unknown mechanism modeled by a species

¹We abbreviate the official name of the BioModels by reducing the number to 3 digits. The original name of the model is *BIOMD000000149*.

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

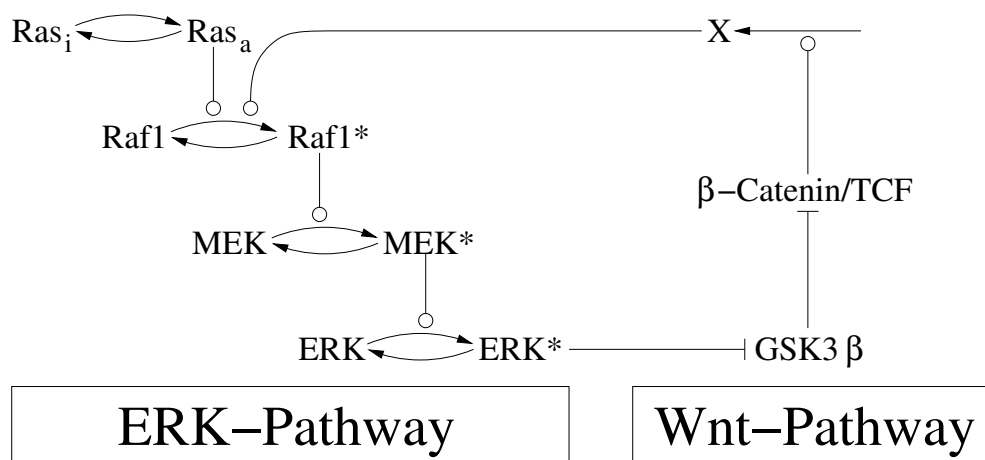


Figure 2.2: Simplified representation of the reaction network from *BIOMD149* (41) combining the ERK- and Wnt-signaling pathways. The Wnt signal, serving as input to both pathways, is not shown. Lines with circles represent essential modifiers identified with the presented approach. Lines ending in orthogonal bars indicate inhibition. Abbreviations: ERK, extracellular signal related-kinase; GSK-3 β , glycogen synthase kinase 3 β ; TCF, T-cell factor.

called “molecule X”. The transcription of this molecule is modeled to be up-regulated by a complex of β -catenin and TCF. The availability of β -catenin is regulated by active GSK-3 β , which in turn is inactivated by phosphorylated ERK. According to the model, X up-regulates the signaling through the ERK-pathway. The rates of phosphorylation of the different levels of the ERK-pathway are modeled with kinetic laws. Thus, a high concentration of phosphorylated Raf increases the rate of phosphorylation of MEK, which in turn increases the rate of phosphorylation of ERK.

Without the processing of the kinetic laws the network contains 384 reactive organizations generated from the union of 11 elementary organizations. After processing, the network contains 150 reactive organizations generated from the union of 18 elementary organizations. Thus, the number of reactive organizations declines while the number of elementary organizations increases. Figures 2.3 and 2.4 depict the Hasse diagram of elementary organizations of both networks. The Hasse diagram of the unprocessed network (Figure 2.3) displays a very simple structure. The smallest organization already contains X. From the kinetic law of the production reaction of X it can be determined that a positive concentration of the complex β -catenin/TCF is required for a non-zero flux of this reaction. But this is not taken into account since this constraint is modeled

2.3 Organizational Structure Of The ERK/Wnt-signaling Pathway

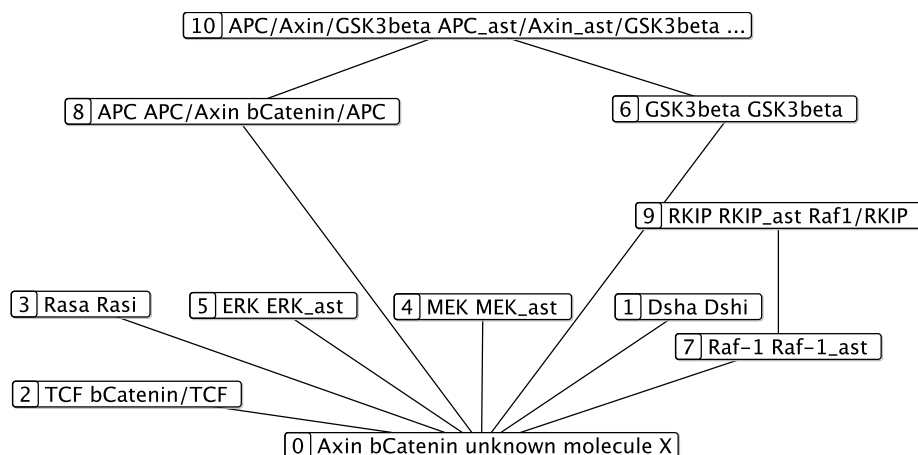


Figure 2.3: Hasse diagram of elementary organizations of *BIOMD149* without processing of the kinetic laws. Only species appearing for the first time in each organization are shown. For example organization 9 contains the species displayed in the nodes corresponding to organization 0, 7, and 9. Not all species in organization 10 are displayed. A list of abbreviations can be found in Supplement 7.4. Phosphorylated forms of a protein are denoted by the suffix “_ast”. Active/Inactive forms by the suffix “a”/“i”.

through the modifiers of the reaction and not on the level of substrates and educts as required by the Feinberg condition. Consequently, the different levels of the ERK-signaling pathway are also present independent of each other. This can be observed by the presence of the corresponding phosphorylated and dephosphorylated proteins directly above the smallest organization in the Hasse diagram.

From a simulation perspective, the reactive organizations of the original network would indicate a state of the network where, for example, MEK and MEK* as well as the input species could be constantly present (Figure 2.3, Organization 4). However, by examining the kinetic laws of the phosphorylation of MEK to MEK* we find that this reaction has a flux of zero if the species Raf1* is not present. Thus, only the dephosphorylation of MEK would have a positive flux, finally using up all MEK*. After processing of the corresponding kinetic law, Raf1* is identified as an essential modifier and added as a catalyst to the reaction, as seen in the Hasse diagram of the processed network (Figure 2.4). The organization containing the species MEK* and MEK (Figure 2.4, Organization 8) is situated above the organization containing Raf1* (Figure 2.4, Organization 6).

From this perspective the processing of the kinetic laws can be seen as adding mech-

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

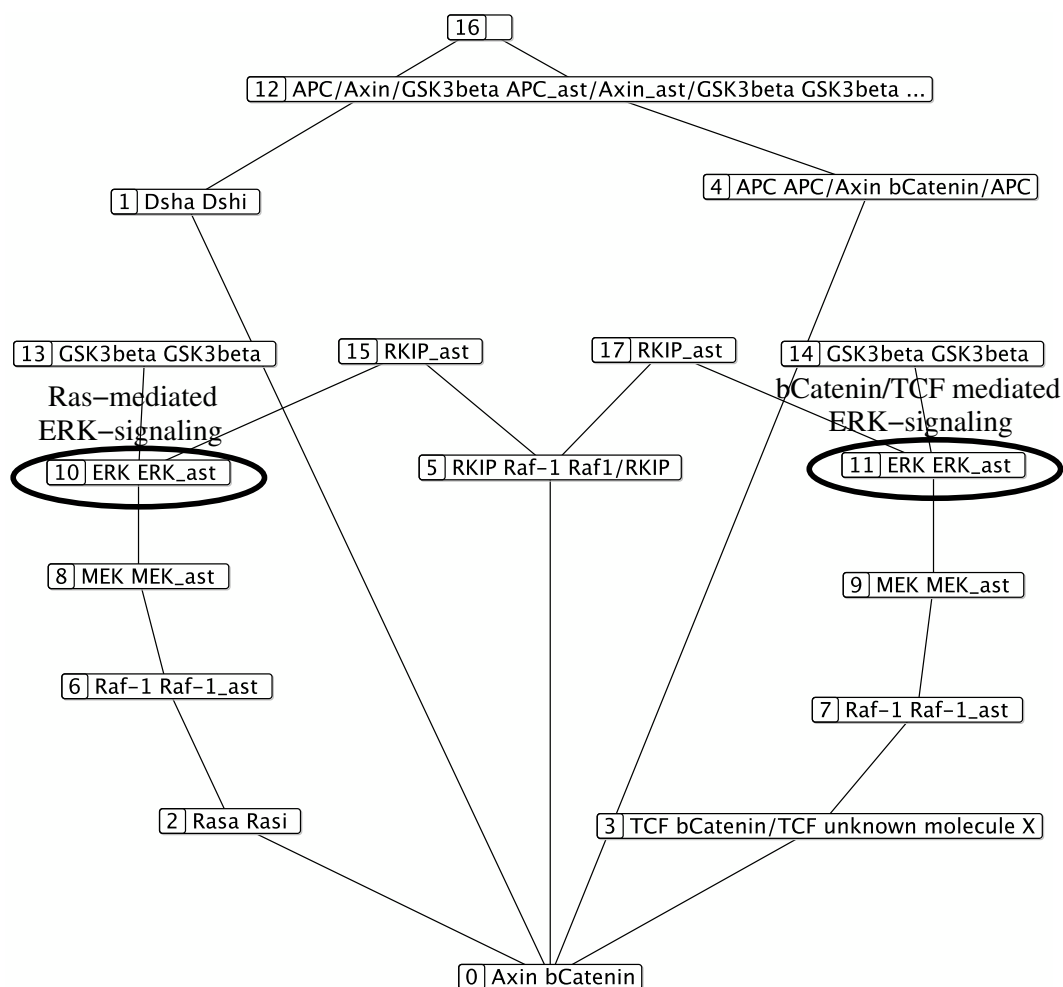


Figure 2.4: Hasse diagram of elementary organizations of *BIOMD149* after the processing of the kinetic laws. Only species appearing for the first time in each organization are shown. Not all species in organization 12 are displayed. Naming follows the same conventions as in Figure 2.3. The different pathways for up-regulation of the ERK-signaling pathway are indicated. In comparison to Figure 2.3 we find, for example, the node corresponding to organization 6 above the node corresponding to organization 2 (corresponding to the nodes labeled 7, respectively, 3 in Figure 2.3). This corresponds to the conclusion that a positive concentration of Rasa and Rasi is required for the presence of Raf1 and Raf1* in the limit behavior. Comparison with Figure 2.3 shows that this conclusion can be drawn only if the kinetic laws are processed.

anistic detail to the reactions. Thus, when we find Raf1* necessary for the phosphorylation of MEK to MEK*, the addition of the modifier Raf1* as catalyst corresponds to the complex formation between Raf1* with MEK prior to phosphorylation. The

approach to consider kinetic laws in OT can be seen as refinement of the reactions of a model making use of the additional information present in kinetic laws. Even though OT does not explicitly require the kinetic laws of a reaction network, knowledge about them can be used to better predict the limit behavior of a reaction network. Conversely, in the sense of the Feinberg condition, the underlying mechanisms are modeled more accurately on the stoichiometric level of the network if this approach is used. In agreement with the results of Kim et al. (41), we find an alternative route for the activation of the ERK-pathway, indicated by the organizations 3, 7, 9, and 11 in Figure 2.4. Through the action of the complex β -catenin/TCF, the transcription of X is up-regulated and, thus, bypasses the activation of Raf by Ras. A constant activation of β -catenin/TCF, for example through a mutation, can result in a decoupling from any signal and consequently lead to a constant up-regulation of the ERK-signaling pathway, as is often found in cancer (41). In the unprocessed network, we do not obtain these results.

2.4 Large-Scale Analysis Of Bio-Models

In order to demonstrate the utility of our approach we analyze the models of the eleventh release¹ of the BioModels database (44). This database contains 185 manually curated models of biological networks in SBML format.

SBML allows species to be defined as external. Thus, their concentration is assumed constant. For the computation of chemical organizations we add an inflow and outflow reaction of the form $\emptyset \rightarrow s$ and $s \rightarrow \emptyset$ for each external species s . For all except 3 models we were able to compute the reactive organizations using the deterministic algorithms for organization computation (see Centler et al. (11) for algorithmic details). For the remaining three models (*BIOMD014*, *BIOMD019*, and *BIOMD049*), a heuristic based on a random walk strategy to determine organizations (11) needed to be applied. Since we wanted to identify species appearing in no organization and each of these models did contain an organization encompassing the entire species set, computation of the complete set of organization was not necessary for these models.

¹The BioModels Database is updated in releases whereby models are corrected or added. We downloaded the models used in this work on 20th October 2008.

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

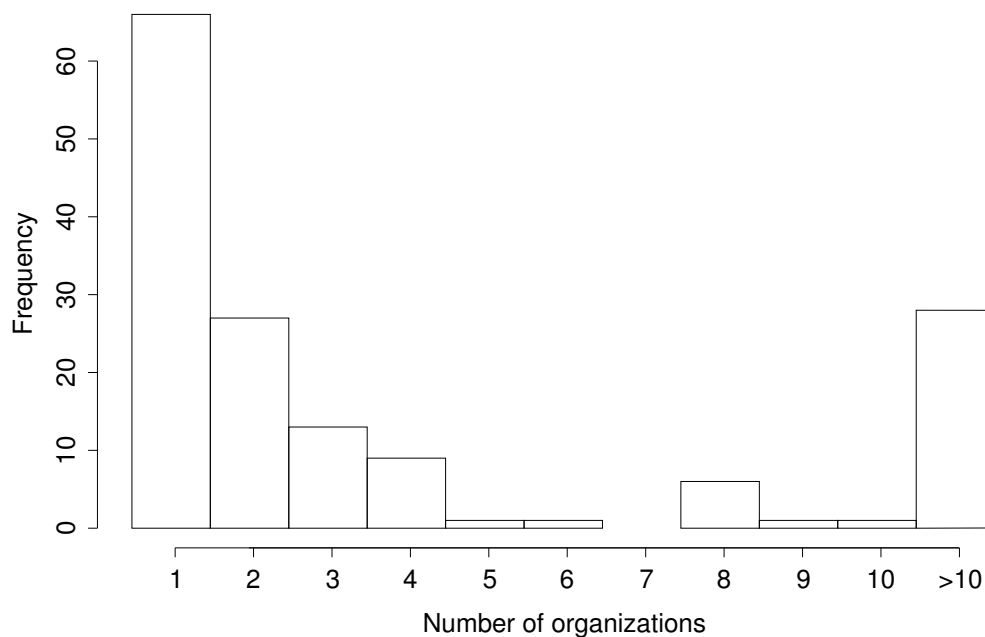


Figure 2.5: Histogram of the number of reactive organizations in the models of the BioModels Database. Please note that this number includes 6 models with more than 1000 organizations (listed below > 10 organizations).

A total of 172 models contained a non-empty organization. In the remaining 13 models, only the empty organization was found since they contained neither reactions nor species. An overview of the number of reactive organizations is given in Table 2 in Supplement 7. While 77 models contained only a single reactive organization, the highest number of organizations was found in *BIOMD175*, with 319,248 reactive organizations. An overview of the distribution of the number of organizations can be found in Figure 2.5.

Species participating in no reaction can drastically increase the number of organizations in a network. Thus, we computed only the reactive organizations in each network and omitted species participating in no reaction (in 24 models) from the analysis. In 31 models some species did not appear in any reactive organization. A first analysis showed that this set contains many models where such behavior was intended. Thus, in several models the concentration of some species was set to a non-zero value at a given time point (e.g., $t = 0$). To take into account this short-time behavior we added an inflow reaction for each such species. Doing this, we found that only 5 models with species absent from any reactive organization remained: *BIOMD044*, *BIOMD093*,

BIOMD094, *BIOMD143* and *BIOMD151* (see Table 2.1). By analyzing the reactions in which the missing species participated and comparing the SBML models to their description in the corresponding publications, we found potential inconsistencies. We identified all these inconsistencies as actual modeling errors.

2.4.1 Resolving Network Inconsistencies

In 3 of the 5 models, *BIOMD093*, *BIOMD094* and *BIOMD143*, we identified reactions that were set to irreversible despite their kinetic laws producing negative fluxes in the course of the simulation, as described in the corresponding publications. Thus, they were indeed reversible and we modified them accordingly. Repeating the analysis, we found all species present in the reactive organizations of *BIOMD093*. In *BIOMD094*, missing species remained. However, this was an intended behavior since a gene knock-out was modeled (68).

In *BIOMD143* we still found some species absent after we had changed reactions with negative fluxes in the simulation to be reversible. This model describes the oscillatory metabolism of activated neutrophils (50). A simplified and decompartmentalized version of the relevant reactions is depicted in Figure 2.6. The species absent from the reactive organizations are hydrogen (H_2^+) from cytoplasm and phagosome. The model contains only reactions consuming these two species. The simulation of the ODEs even produces negative concentrations of both. The reason for the consumption of these species is inconsistent modeling of the stoichiometry of the reactions and an inconsistent kinetic law. Cytoplasmatic and phagosomal hydrogen are consumed together with superoxide (O_2^-) to produce hydrogen peroxide (H_2O_2). In the course of the disposal of H_2O_2 by ferric peroxidase in the phagosome, an additional 4 protons from melatonin (MLTH) are consumed to produce the initial form of ferric peroxidase. With the exception of ferric peroxidase and free radicals of melatonin (MLT), all species are consumed without producing equivalent products. Thus, the disposal of H_2O_2 by ferric peroxidase consumes oxygen and protons. The model contains an inflow for NADPH and O_2 . Oxidation of NADPH by oxygen or free radicals of melatonin can produce superoxide and melatonin respectively. Thus, there is a constant inflow of NADPH and oxygen that can replenish the consumed species. However, the kinetic law of the production of superoxide from O_2^- and hydrogen does not depend on the concentration of hydrogen in the model. Together with a zero initial concentration of hydrogen, the

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

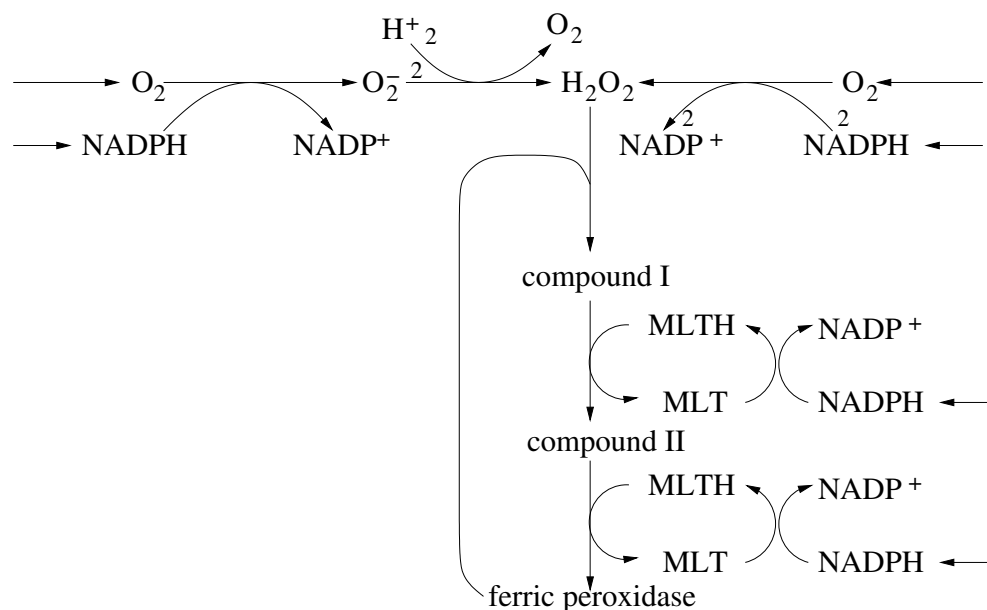


Figure 2.6: Simplified representation of the reactions of *BIOMD143*. As a result of inconsistent stoichiometries hydrogen and oxygen are consumed in the course of detoxification of hydrogen peroxide. There is only an inflow of oxygen, and the consumption of hydrogen does not depend upon its concentration. Consequently a simulation leads to a negative concentration of hydrogen. Abbreviations: MLTH, melatonin; MLT, melatonin free radical

simulation of the model leads to a negative concentration of this species. Making the rate law dependent on the concentration of hydrogen resolves the problem of negative concentration of hydrogen. Additionally, either removing the inconsistencies in the stoichiometry or adding an inflow for hydrogen allows positive concentrations of this species during simulation.

The reasons for the missing species in *BIOMD044* are very similar. Here a species is modeled to serve as a pseudo-substrate to a reaction that could have been modeled without substrate. The kinetic law governing the reaction does not depend upon the concentration of this substrate. Since it is not produced by any other reaction, negative concentrations appear in the course of the simulation. Replacing the respective reaction by an inflow reaction resolves the problem.

In *BIOMD151* almost all species are absent from reactive organizations. This network represents an integrated model of the JAK/STAT and ERK-signaling pathways regulated by IL-6 in hepatocytes (61). A detailed analysis of the model and the set of ordinary differential equations presented in Singh et al. (61) showed that a complex for-

2.4 Large-Scale Analysis Of Bio-Models

Model	Description	Species/ Reactions	Reactive Orgs.	First Step		Second Step	
				OT (spec./rea.)	FBM (rea.)	OT (spec./rea.)	FBM (rea.)
BIOMD037	Sporulation control network in <i>P. polycephalum</i> (24)	12/14(14)	1(2)	3/6	10	12/14	14
BIOMD044	Model of intracellular calcium oscillations (6)	7/8(8)	2(2)	3/4	5	6/7	7
BIOMD093	JAK/STAT signal transduction pathway (68)	34/48(48)	5(3)	11/16	30	31/43	43
BIOMD094	JAK/STAT signal transduction pathway (68)	34/47(47)	2(3)	5/5	27	24/24	40
BIOMD143	Oscillatory metabolism of activated neutrophils (50)	20/20(20)	1(1)	4/4	4	7/5	5
BIOMD149	Crosstalk between Wnt and ERK Pathways (41)	28/39(39)	150(384)	28/39	39	-	-
BIOMD151	IL-6 signal transduction in hepatocytes (61)	68/114(114)	80(96)	49/71	111	19/14	112

Table 2.1: Selected results from the large-scale analysis. See Supplement 7 for the entire table. The 5 models in which inconsistencies have been identified are shaded in light gray. The first 4 columns give general details about the models. Numbers in brackets indicate the number of reactions of the original network that can increase through processing of the kinetic laws. The number of species remains constant. The fourth column gives the number of reactive organizations in the modified and (in brackets) the original network. In the fifth and sixth columns species and reactions that can be present in the limit behavior of the processed network are given. OT denotes the prediction by organization theory, and FBM the predictions by flux-based methods. In some cases FBM identify more reactions to be present in the limit behavior than OT. These cases are shaded in dark gray. The seventh and eighth columns give the same numbers when inflow reactions for species with an event setting their concentration to a positive value at a certain time-point are added. In cases where the original network already contained all species, those numbers are omitted.

mation step was missing, such that the signal from IL-6 could not be transmitted to the subsequent signaling pathways. Only the complex dissociation reaction was present. During simulation it had a negative flux, mimicking the complex formation reaction. Adding the missing step produced a model in which all species appeared in a reactive organization.

2.4.2 Comparison With Flux-based Methods

Next, we will compare our results with those obtained with flux-based methods, including flux balance analysis (FBA) (65), elementary mode analysis (59), and extreme pathway analysis (58). These methods can be used to check whether a certain reaction can be present in a steady-state flux obeying the irreversibility constraint. Thus, they

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

can predict whether a reaction can be present in the limit behavior of a reaction network. In FBA this can be done directly, while elementary mode analysis and extreme pathway analysis return a set of vectors spanning the solution space of the steady-state condition. However, since OT also takes into account growth states, in which some species accumulate, the steady-state condition is adapted accordingly (details can be found in the Supplement). Furthermore, since we only want to know whether a reaction can appear in any steady state or growth state, we do not need to apply these methods directly, but can use a linear programming approach similar to FBA, outlined in the Supplement.

We compared the predictions of flux-based methods to those of OT for the models of the BioModels Database. With OT we identified 31 models where some reactions did not appear in any reactive organization. The same 31 models are identified using flux-based methods. However, when analyzing the predicted set of available reactions in detail, we found differences in 25 of the 31 models. Due to the definition of self-maintenance, the set of available reactions is a subset of those predicted by flux-based methods. Thus, in all 25 cases, flux-based methods found reactions present in the limit behavior that indeed could not maintain a positive flux in a long-term simulation.

The reason for this difference closely follows a concept presented in Kaleta et al. (35): a steady-state flux in a network uses some species that cannot be produced at a positive rate. In this flux these species might be interconverted into each other or act as catalysts. Further assume that there is a reaction steadily draining some of the unproducible species. Thus, they will finally vanish. In consequence, this steady-state flux cannot be part of any steady state of the complete network. If a particular reaction is present only in such steady-state fluxes, it is predicted to be present in the limit behavior of a reaction network by flux-based methods, while OT correctly identifies it as absent since it correctly takes into account the drain of the unproducible species. We will outline this concept in more detail using *BIOMD037*, a model of the sporulation control network in *Physarum polycephalum* by Marwan (46) (Figure 2.7). While OT predicts 8 of the 12 reactions to be absent from the limit behavior (Figure 2.7 A), flux-based methods identify only 4 such reactions (Figure 2.7 B). The differentially predicted reactions account for the interconversion of Pfr to Pr and Xi to Xa. Flux-based methods find a flux where the conversion of Pfr to Pr and vice-versa is in equilibrium. However, this does not take into account that there is also a reaction irreversibly converting Pr

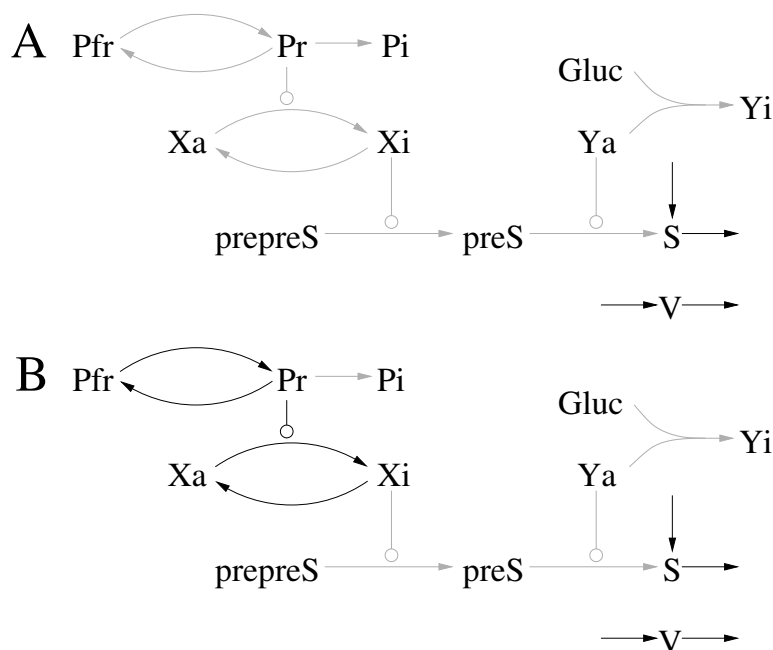


Figure 2.7: Reaction network from *BIOMD037* modeling the sporulation control in *Physarum polycephalum* from (46). Lines ending in circles indicate essential modifiers for a reaction. Light gray reactions cannot have a positive flux in the limit behavior, according to **A** OT and **B** flux-based methods. Abbreviations: Pr, active photoreceptor; Pi, inactive photoreceptor; (pre/prepre)S, sporulation signal (and precursors); Ya/i, active/inactive glucose receptor; Gluc, glucose; Xa/i, active/inactive signal transducer.

to Pi. Thus, a non-zero concentration of Pr will be depleted by the conversion into Pi. In consequence, there is no reactive organization containing Pfr and Pi.

Additionally, we find an interesting case in the interconversion of Xa to Xi and vice-versa. The conversion of Xi to Xa requires the presence of Pr. Flux-based methods identify an equal flux of both reactions as a feasible flux, since Pr acts only as a catalyst. However, the analysis using OT shows that such a flux also requires the presence of Pr. Thus, both species cannot persist in the limit behavior since Pr, required for the reaction of Xi to Xa, will vanish over time. Since Xa is steadily converted to Xi, only this species would finally remain. This demonstrates how our approach takes the kinetic laws into account which is not possible using flux-based methods.

In 2 of the models in which we identified inconsistencies, *BIOMD094* and *BIOMD151*, predictions for the presence of reactions in the limit behavior between OT and flux-based methods differ. In *BIOMD151*, OT predicts 9 reactions to be present, while

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

flux-based methods identify 112 of the 114 overall reactions. As outlined above, flux-based methods can predict only the same or a larger set of reactions to be present in the limit behavior. Thus, the search for inconsistencies is simplified by reducing the size of the system to analyze if OT is used. This is also corroborated by 3 models in the uncurated branch of the BioModels Database containing inconsistencies. In all 3 models, flux-based methods predict more reactions to be present in the limit behavior than OT (see Supplement for further details).

2.5 Conclusion

In this work we demonstrated that information hidden in kinetic laws affects the results obtained from chemical organization theory (OT). We presented an approach that is able to uncover this information. This approach enabled us to refine the chemical organizations in 41 of the 185 models (22%) of the BioModels Database.

The Hasse diagram of organizations of the processed model of a combined ERK/Wnt-signaling pathway took into account the different levels of phosphorylation in the signaling cascade, while the set of organizations of the unprocessed network did not. Furthermore, the Hasse diagram of organizations demonstrated several possible pathways for constant up-regulation of this pathway, an important event in carcinogenesis consistent with the results of Kim et al. (41).

Analyzing the 185 models of the BioModels Database, we checked the behavior of the models during long-term simulation (limit behavior). Thus, we found 31 models where several species could not persist in a long-term simulation. Furthermore, we identified 5 models in which some species could not be present at all during simulation. This was due to inconsistent reversibility constraints in two models, negative concentrations of some species during simulation in another two models and a missing reaction in the fifth model. In the non-curated branch of the BioModels Database we identified 3 models with modeling errors. Comparing the set of species present and the reactions having a non-zero flux in the limit behavior, we found OT able to predict those sets more accurately in 25 models (14%) compared with flux-based methods like flux balance analysis, elementary mode analysis, and extreme pathway analysis. These models account for 81% of the models in which the set of species and reactions present in the limit behavior of the model did not encompass the entire set of species

and reactions. In five of the 8 models of both branches of the BioModels Database in which we detected modeling errors, OT made more accurate predictions in comparison to flux-based methods.

These results demonstrate that OT is a valuable tool in 3 important aspects of network design and analysis. First, when this approach is used to extract additional information from the kinetic laws of the reactions, the set of organizations corresponds to the potential steady state and growth states of a reaction network. Thus, important information about the dynamic structure of a reaction network can be uncovered. Second, OT can be used in an iterative fashion to assist in model building by identifying inconsistencies that need to be resolved. Third, OT more faithfully identifies parts of a network whose maintenance is not yet explained than flux-based methods. Thus, it is of particular interest for identifying gaps due to missing knowledge in large-scale metabolic networks as documented in Centler et al. (11). In consequence, it can be beneficial for methods aiming to remove such inconsistencies (43, 54). In the other direction, our approach could be extended by these methods to automatically propose changes in order to remove inconsistencies. However, computational constraints currently prohibit the application of our deterministic algorithm to very large networks (e.g., more than 500 reactions). An approximation can be used for networks of this size, but the results require manual checking. A more efficient algorithm that will enable the application of OT to genome-scale networks is in development.

For a more detailed outline of the processing of the kinetic laws, see the Supplement (7.3.3.4).

2. CHECKING SBML BIOMODELS USING HIDDEN DATA

3

NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

3.1 Introduction

The chemical industry has become a very important economical factor in the last centuries. Unfortunately, ecological awareness arose very late during the development of this scientific field and its applications. Thus, today we have to cope with the bothersome inheritance of closed down factories, brownfield sites and contaminated grounds. For a long time, dumping was the only option to deal with such contaminated soil; apparently cleaning up the old industrial sites, but in no way eliminating the pollutants left in it. What is more, these dumps even concentrate hazardous substances and thus pose a huge threat to groundwater and diverse ecosystems. Fortunately, microbiological and systemic insights gained in the last decades open up new perspectives to tackle these issues: at least a subset of the possibly dangerous chemicals might be dealt with by microorganisms which have evolved to be capable of dealing with contaminated environments. This means either that

- the respective organisms are able to withstand the toxic effects of the substances,
- the species are able to (at least partially) catabolize the chemicals, thereby modifying their own environment, or

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

- they are even able to profit from energy stored in the compounds.

This, in turn, led to the idea that biochemical systems might be engineered to facilitate the microbiological decomposition of contaminants in soils, thus offering an option to actually cleanse our inherited liabilities (16, 63).

3.2 Preconditions For Multi-species Network Models

While it might be technically feasible to engineer genetically modified organisms to optimally reduce the amount of pollutants in certain soils, a less intrusive treatment would be to combine well-known organisms in such a way, as to allow for a full mineralization of the harmful chemicals in a given environment based on their concerted natural activities. Of course, *in vitro* it would be possible to just combine organisms randomly and observe the outcome to find possible candidate sets for *in situ* application. Yet this trial-and-error approach would be rather tedious and most probably not fitted to reach optimal degradation rates. A more scientific approach uses prior knowledge about the location of interest and insights collected in biochemical databases.

It is well known that virtually every organism has preferred environmental conditions allowing sustainable growth. These conditions are shaped by the following factors given in the (not necessary complete) list:

- aerobic/anaerobic environment
- pH value
- salinity
- temperature
- presence of heavy metal ions
- presence of electron acceptors
- microorganisms present in the environment
- pressure
- present pollutant substances

3.3 Basic Approach Description

Omitting the possibility of tailoring organisms by means of genetics, the ultimate goal to cope with the aforementioned biodegradation problem would be the following: A tool that predicts, for a given environmental situation, which set of bacteria could be applied and which additional substances might have to be added. To achieve this, the following steps have to be performed:

1. find bacteria able to process the objective substances.
2. filter bacteria by feasibility with respect to the environmental situation.
3. find a flow distribution optimized for pollutant degradation.
4. determine which substances have to be added to allow this optimal flow.
5. find degradation intermediates released by the utilized set of bacteria.
6. find bacteria able to process the intermediates.
7. repeat from step 2 until no harmful chemicals are left unprocessed.

3.3.1 Search For Processing Bacteria

This step can be accomplished in the following way:

1. search online databases for reactions processing the substances of interest. In other words: find reactions that have the objective substance in their list of educts in any of the feasible directions.
2. search for enzymes able to catalyze the respective reactions.
3. search for organisms whose genomes code for the found enzymes.

Although the presence of a gene coding for an enzyme does not automatically imply the presence of the enzyme, this is a good estimate to find a set of potential degraders.

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

3.3.2 Filter Bacteria

When a set of potential degraders has been found, it is advisable to get rid off all species which are not compatible with the preset environment. For example, these might be obligate anaerobe bacteria which are not feasible for an aerobic environment, or species that will not grow at the present pH value. This filtering demands for databases which hold descriptive data for listed organisms.

3.3.3 Optimize Degradation And Determination Of Additional

After the previous steps, we should have found bacteria that potentially have the capability to break down a substance of interest. The next step is to find out under which conditions this degradation can reach its highest performance. Therefore, the network of relevant bacteria has to be fed with the nutrients present in the respective environment while trying to maximize the inflow of the substances of interest. Alongside this optimization, we will possibly find a set of chemicals that are not yet present in the cell's environment but can be added to allow for, or at least speed up, the process.

3.3.4 Find Secreted Intermediate Substances

This is a crucial step during the construction of interaction networks of bacteria, since a singular species might not be capable of a full mineralization of a given pollutant. In this case, it is very likely that intermediate substances, i.e. the products of the partial degradation, will be activeley or passiveley exported by the bacteria and will thus accumulate in the environment. As long as those substances pose a danger to the environment or even inhibit the growth of their producers, there is a need to get rid of these products. There are several approaches towards the solution of this sub-task, like the detection of metabolic interfaces (4) and variances of the FBA (51) or the seed set method (5). The general procedure would be the following: for each organism found in the previous steps determine the substances produced in a "healthy" (standard) environment while optimising biomass production. Compare those metabolic products with the substances produced when the standard environment is altered by forcing the organism to process the pollutant substances. This should yield a set of additionally produced substances, some of which might be problematic as well. A major obstacle here is the question how to determine whether a particular product is

problematic. Usually, one would refer to scientific literature to clarify this. Also, there are approaches to decide this question based on structural properties of the respective molecules. Furthermore, it could be decided by the location of the substance within the metabolic network of an organism: while useful substances seem to appear in the highly connected centre of metabolic networks, harmful chemicals tend to appear in the periphery (53). For an automated approach this knowledge could again be harvested from online databases.

3.3.5 Iterative Search For Further Degradation Pathways

At this point we should have found bacteria which are potentially able to degrade our substances of interest to a certain point and possibly leave us with a set of degradation products that are not fully mineralized. As the ultimate purpose was a set of bacteria with the potential to fully degrade the raw pollutant mixture, we have to iterate the previous steps and need to identify bacteria able to degrade the intermediate products to a satisfactory set of tolerable output substances. Thus, the aforementioned steps have to be executed recursively. Ideally, with the help of these repeated steps we would be able to determine sets of bacteria with the potential to completely eliminate the contaminants.

3.3.6 Finding The Optimal Set Of Bacteria

If several sets of bacteria were found by this method, it would be reasonable to construct integrated metabolic network models of each of those sets of bacteria. Those models should comprise the full degradation path and allow for qualitative and quantitative comparison of the respective bacterial communities: it is advisable to review the stoichiometric side conditions for the degradation path and check whether a net flux through this path is feasible under the given environmental conditions. Complementary to these predicted pathways and preconditions, it is necessary to perform *in vitro* and *in situ* experiments to prove the possibility to actually perform the complete degradation, while the concentration of intermediates has to be monitored.

3.4 Implementation

To automatize the data-driven steps of the preceding description, we wanted to implement a toolbox allowing to enter environmental parameters and search for potential degraders. For the fast operation of this toolbox, all data used needed to be stored in a local database, as online database requests would inadmissibly slow down any operation. Also, we wanted the design to allow several calculations at the same time, thus having it follow a client-server architecture. For easy portability to other operating systems, all software was implemented in Java and can be downloaded from <https://github.com/FSU-Jena/InteractionTools>.

3.4.1 Client-Server Architecture

Since we wanted to allow to process several tasks in parallel, we needed to run the calculations in independent threads. To provide even more computational power, these threads had to be moved to separate machines, while remaining under central control. Thus, the whole toolbox was divided into four logical units which communicate with each other:

1. the graphical user interface allows to define tasks and browse results.
2. the server module accepts defined tasks, schedules their execution and gathers the results.
3. the client modules accept scheduled tasks, do the calculations and hand back the results to the server module.
4. the database module is accessed by all other modules in order to provide necessary data for task creation and execution.

Figure 3.1 gives an overview on these units and how they are connected.

3.4.2 The Local Database

3.4.2.1 Structure Of The Local Database

This is a central part of the software package, as the DB feeds the other modules with all kinds of data. The database itself was designed to reflect the needs of the software

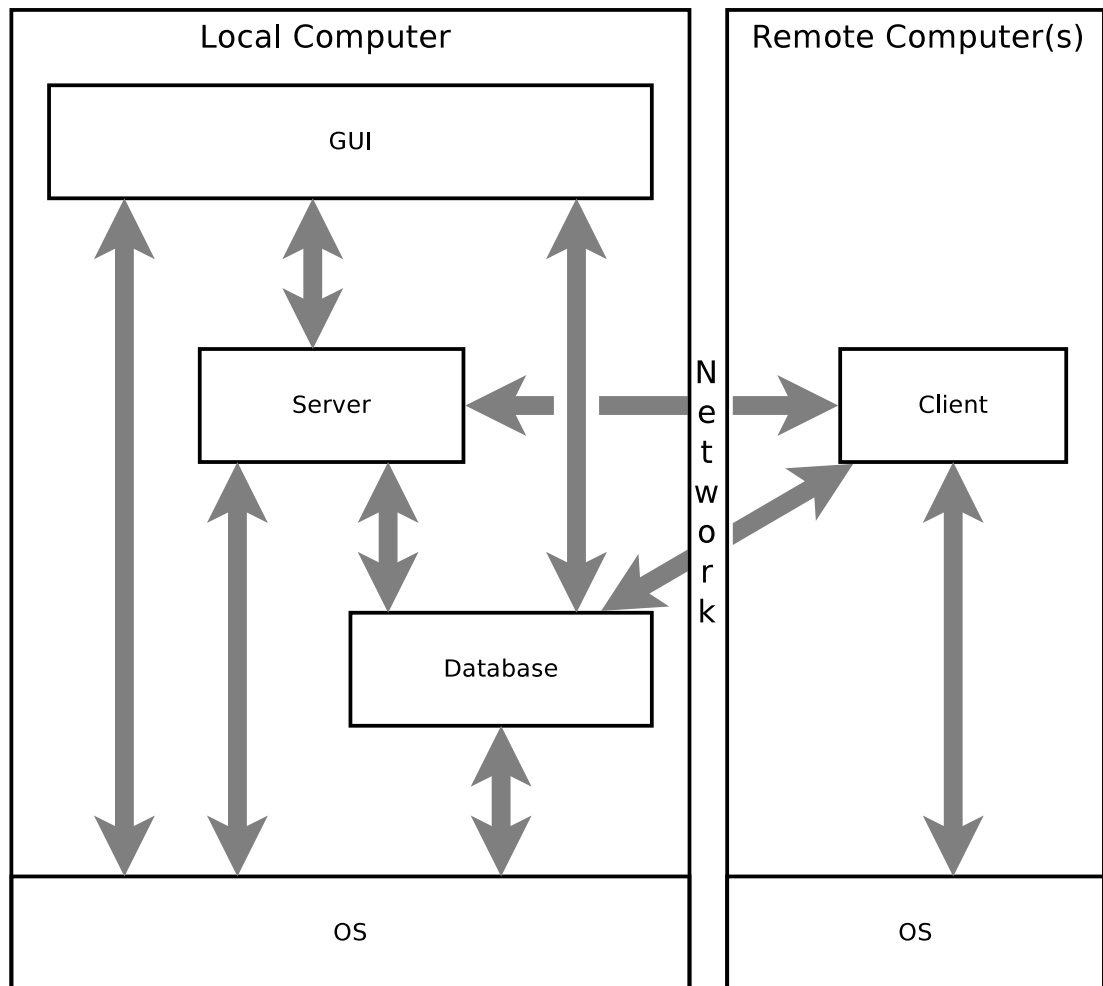


Figure 3.1: The logical subunits of the interaction toolbox. The GUI allows to create tasks and displays their results. Tasks are dispatched to the clients via the server module, clients do the calculation and hand back the results. All modules have reading access on the database.

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

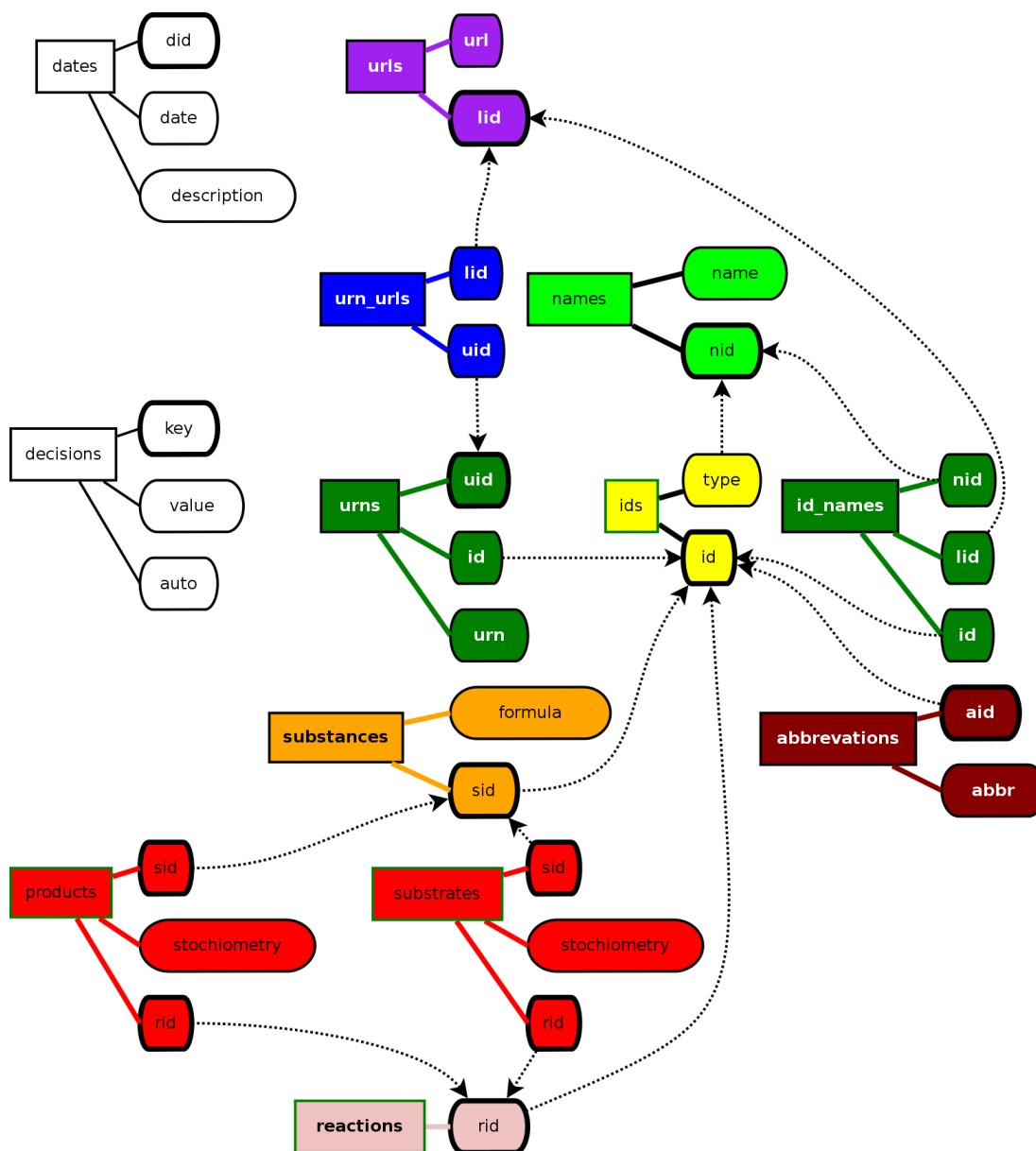


Figure 3.2: Structure of the Local Database.

Tables (rectangles) consist of fields (round boxes) related by continuous lines. Dotted arrows indicate references between fields in different tables.

to be built upon it. The main tables are listed in the following. The data fields are denominated in brackets, primary keys are underlined.

- substances (id, formula) - storing formulas of substances
- enzymes (id, ec, substance) - stores ids of enzymes, their EC-number (if given) and the identic substance (where applicable).
- reactions (rid, spontan) - storing spontaneity information for reactions
- substrates and products (sid, rid, stoich) - holding references about which substances take part in which reactions and their respective quantity
- reaction-enzymes (rid, eid) - linking reactions to the enzymes by which they are catalyzed
- reaction-directions (rid, cid, forward, backward) - holding flags that determine whether a reaction has been marked as directional in a certain compartment by any source
- compartments (id, groups) - a list of compartments and organisms from the harvested databases. The groups field is a reference to a name of a group the compartments belong to, for example “Eukaryotes”
- enzymes-compartments (eid, cid) - this table links enzymes to compartments, i.e. it states which enzymes occur in each compartment.
- hierarchy (container, contained) - storing information about which compartment resides inside of another

Additionally, the database comprises some meta-information tables. These meta-data tables capture information on the origin of the other data; which becomes important if inconsistencies have to be traced:

- ids (id, type) - manages the ids for all aforementioned data sets and states to which type of object the id belongs.
- names (nid, name) - manages all names used for entities within the database.
- urls (lid, url) - collects URLs referenced by various entries.

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

- urns (uid, id, urn) - collects URNs referenced by virtually all entries.
- id_names (id ,nid, lid) - links any object (by id) with its names (by nid), also stores the id of the URL of the page on which the name was declared.
- id_ranges (nid, min, max) - tells which range of ids belongs to data from which database.
- urn_urls (uid, lid) - stores which URN was provided by which webpage.
- abbreviations (id, abbr) - stores abbreviations for certain database entities used on different websites.
- dates (did,date, description) - a log written when the database is filled.
- decisions (keyphrase, value, autogenerated) - saves decisions made throughout the database filling process and whether a decision was based on a heuristic or on user input.

The full DB sheme is depicted in Figure 3.2. We are using MySQL 5 as our database system, which is running in the userspace of a debian wheezy distribution.

3.4.2.2 Data Collection

The database was designed to collect and interweave data from multiple sources. At the time the first tests were carried out, it comprised entries from the following sources:

- Kyoto Encyclopedia of Genes and Genomes (KEGG, c.f. 4.3.1)(39),
- the BioModels Database (45),
- the BIGG database (57),
- The Genome-Scale Metabolic Network Database¹,
- cyanobacteria models provided by the workgroup of Metabolic Network Analysis group of Dr. Ralf Steuer, Berlin,
- some simple testcases to inspect the function of the calculation routines.

¹<http://synbio.tju.edu.cn/GSMNDB/gsmndb.htm>

A declared goal of the database was to allow to combine models from different sources. For this purpose, the substance-datapages were crawled for URN annotations: If the program that fills the database encounters two substance entries with the same URN assigned to them, it checks whether the chemical formulas given for the entities are compatible and writes a common database entry joining names, formulas, abbreviations and URNs. If the formulas of two datasets with equal URNs are mismatching, a dialog box is presented to the user, giving him three options:

1. Unite the entries using the formula already in the database.
2. Unite the entries using the chemical formula from the new source.
3. Keep the respective entries separated.

In this manner, for each of the datasources, the substance list, the list of enzymes, and the list of reactions are read and integrated into the local database.

3.4.3 Graphical User Interface

This is the frontend of the toolbox presented to the user. It is organized in tabs, which group elements by functions. There is one tab for each of the following functional groups:

1. tasks
2. results
3. network view
4. database
5. information.

The *Tasks* tab is further divided into a box containing task buttons and a panel with several lists. This lists panel is subdivided in a form where species can be selected, a form where substances can be selected and a parameters form. Note that within the InteractionToolbox, the term **species** refers to a biological species and can be either a single compartment model or a group of nested compartments. The species tab (Figure 3.3) contains two lists: one provides the list of species available in the

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

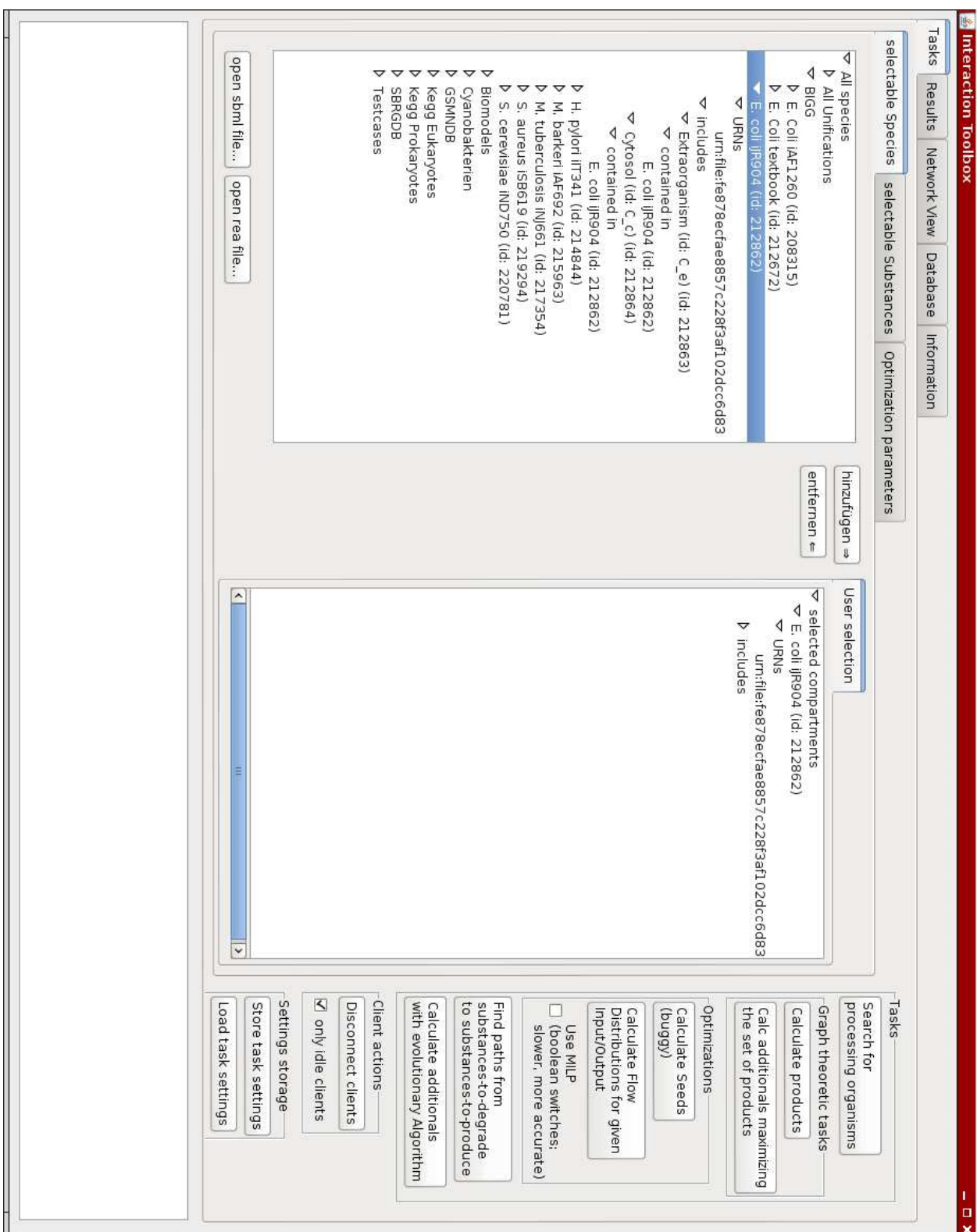


Figure 3.3: The species lists form.

local database grouped by their origin, the other list is the collection of user-selected organisms used for calculations. The second tab within the *Tasks* tab is the *Substances* tab (Figure 3.4), which also contains a list of available substances and four lists of user-selected substances. Within the InteractionToolbox, drugs, glycans as well as chemical compounds and other chemical entities are subsumed to **substances**. Each of the four user lists can be assigned to one of the four properties “substances to degrade”, “substances to produce”, “substances to ignore” and “forbid outflow”. These lists provide the following functionality:

- The “substances to degrade” list is used when degraders for a certain set of substances are searched or the products of a certain organism are calculated.
- The “substances to produce” are used in optimizations where a certain output set of substances is desired.
- Substances in the “substances to ignore” list are considered buffered or in excess, i.e. optimization methods do not try to balance their concentrations. Normally, water would be added to this list.
- Substances in the “forbid outflow” list are forced to be both balanced and without outflow. This aims to find paths avoiding certain substances in the product set.

The third form within the *Tasks* tab is the *Optimization Parameters* form (Figure 3.5), where constants for weighted inflow and outflow terms and the importance of the minimization of the total number of reactions can be set up. The aforementioned buttons box groups buttons used to trigger the actual optimization routines (described subsequently to the GUI description).

The second main tab is the *Results* tab which only contains one form in which all results are displayed in a browsable tree structure (Figure 3.6). It is neighboring the *Network View* tab that can display selected substances and their related reactions within the user-selected organisms (Figure 3.7). The fourth tab is the *Database* tab (Figure 3.8). This tab has two major functions: first, it allows to add new SBML models to the database. Second, it provides a list of all substances in the database whose data have been compiled from several sources. Clicking on one of these substances toggles an analysis, which calculates the probability for each pair of sources that the merged entries really belong together. This is established by comparing names, formulas and SMILES

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

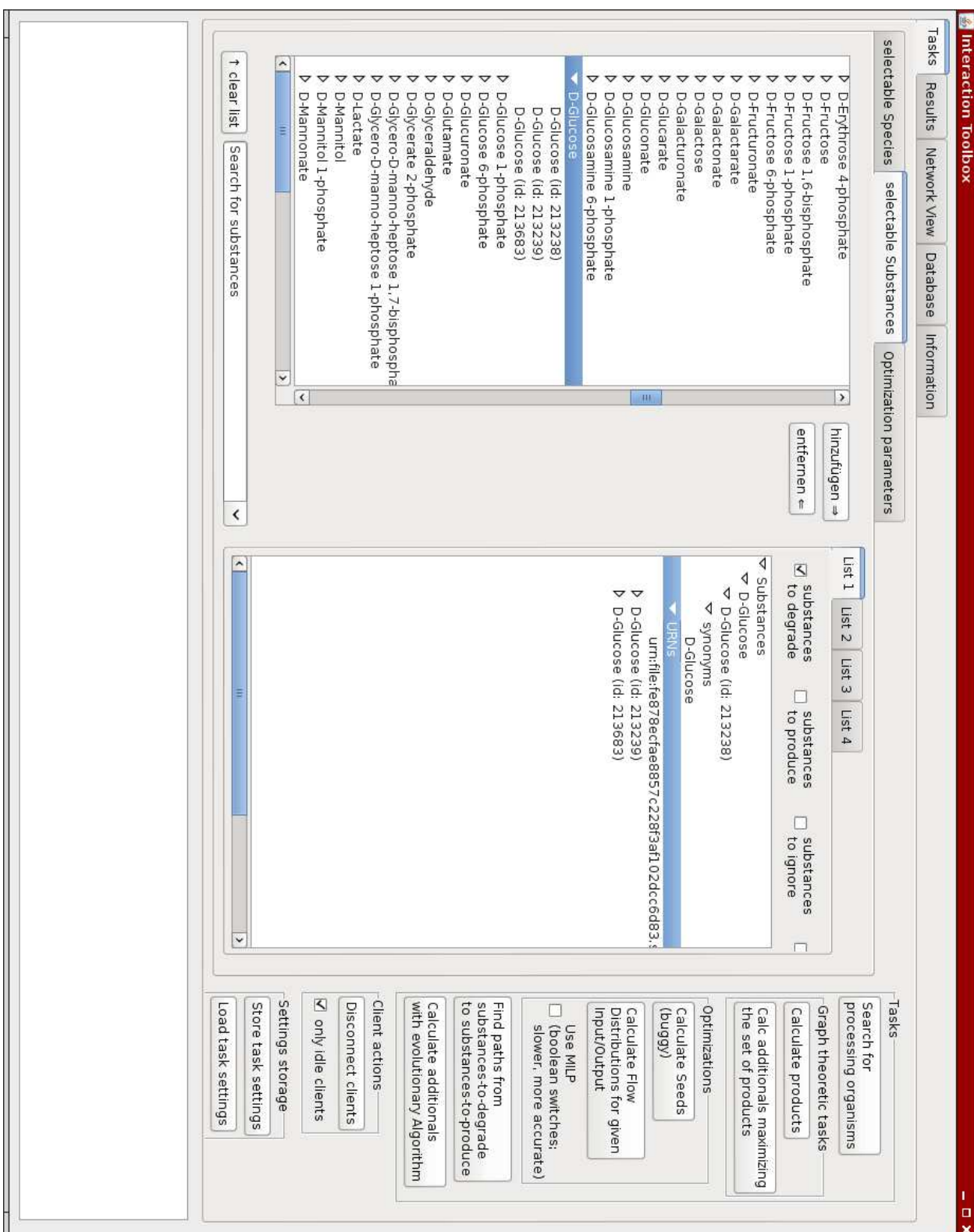


Figure 3.4: The substances lists.

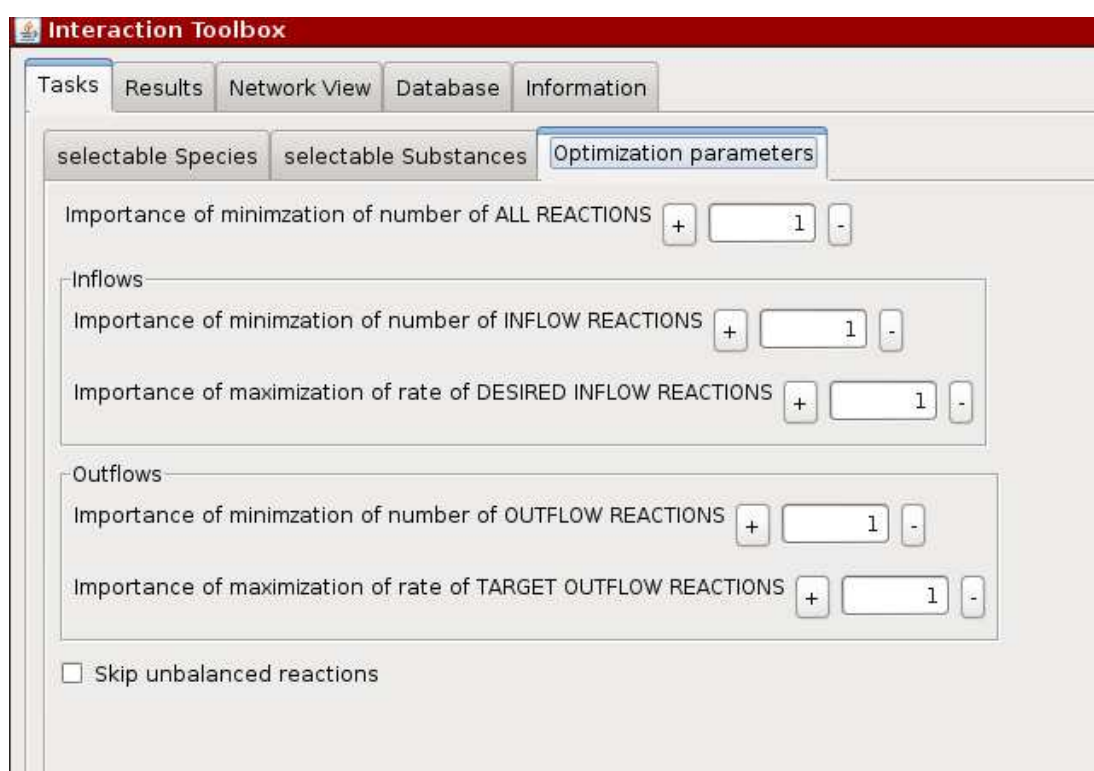


Figure 3.5: The optimization parameters tab.

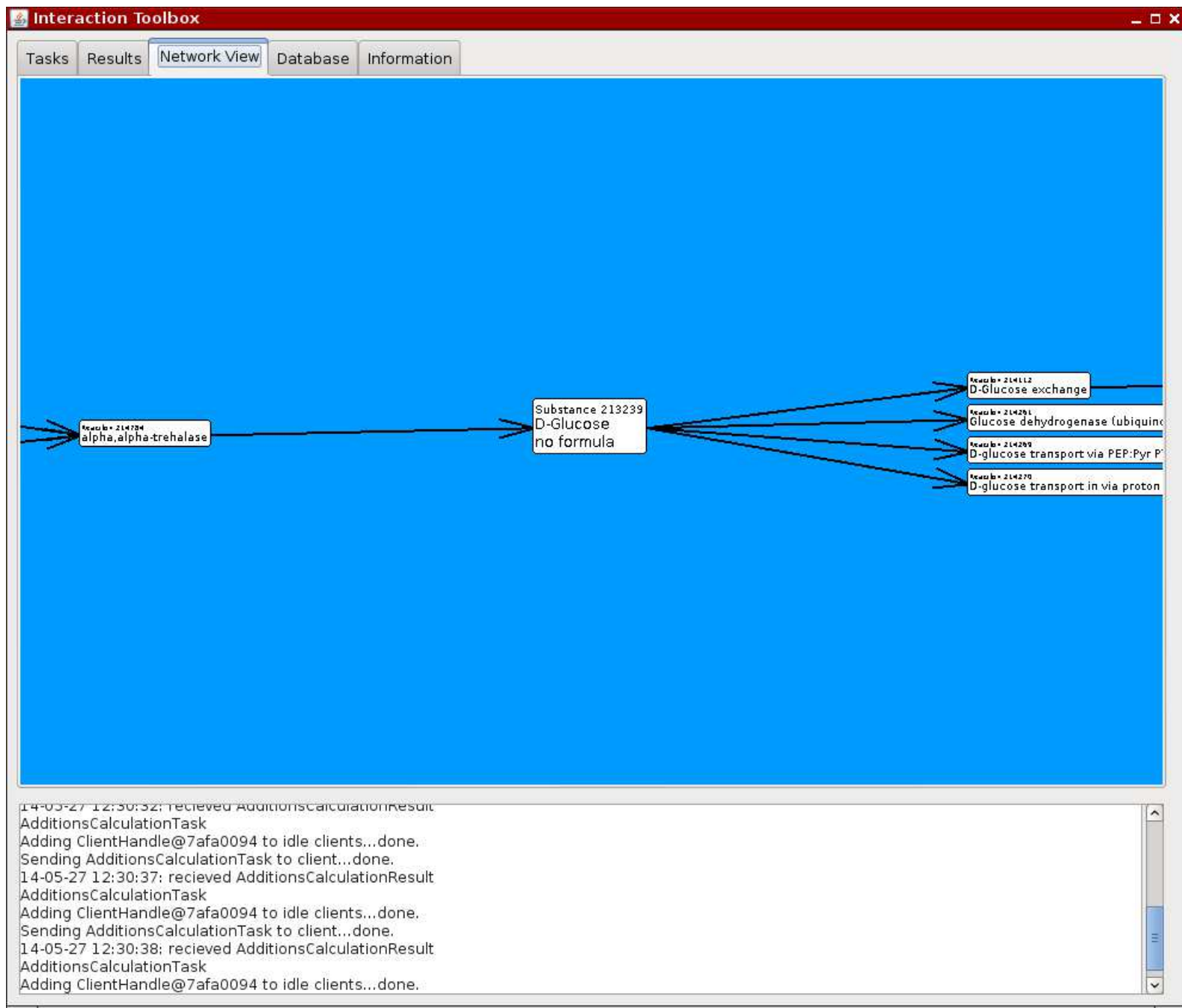


Figure 3.7: The network view - showing the reactions related to D-Glucose in an E.Coli model

The screenshot shows the 'Database' tab in the Interaction Toolbox. The interface includes a 'Tasks' menu with options like 'load SBML file', 'Kegg Eukaryotes', and 'clean Database'. A list of substances is displayed, with 'D-Lysine' selected and its 'similarities of urns' expanded. The list shows various URNs and their sources, such as 'urn:miriam:jcsd:J9.203G' and 'urn:miriam:obo.chebi:CHEBI:16855'. A network diagram on the right shows nodes 1 through 8 connected by lines, with a red dashed line and yellow dashed lines highlighting specific connections. A legend below the diagram lists the URNs for each node.

1: urn:miriam:3dmet:B01320
 2: urn:miriam:cas:923-27-3
 3: urn:miriam:jcsd:J9.203G
 4: urn:miriam:kegg.compound:C00739
 5: urn:miriam:obo.chebi:CHEBI:16855
 6: urn:miriam:pubchem.substance:4002
 7: dbget-bin/www_bget?C00739
 8: GSMN056 - Staphylococcus aureus N315/rea.xml

Downloaded from http://www.3dmet.dna.amc.go.jp/cgi/show_data.php?acc=B01320... done.
 Downloading <http://commonchemistry.org/ChemicalDetail.aspx?ref=923-27-3...> done.
 Downloading <http://www.ncbi.nlm.nih.gov/sites/entrez?db=pccompound&term=923-27-3...> done.
 Downloading http://nikkajweb.jst.go.jp/nikkaji_web/pages/top_e.jsp?CONTENT=syosai&SN=J9.203G... done.
 Downloading <http://www.kegg.jp/entry/C00739...> done.
 Downloading <http://purl.bioontology.org/ontology/CHEBI/CHEBI:16855...> done.
 Downloading <http://www.ebi.ac.uk/chebi/searchId.do?chebiId=CHEBI:16855...> done.
 Note: Smiles found: NCCCC[C@@H](N)C(O)=O on <http://www.ebi.ac.uk/chebi/searchId.do?chebiId=CHEBI:16855>
 Downloading <http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?sid=4002...> done.
 Downloading <http://www.ebi.ac.uk/ontology-lookup/?termId=CHEBI:16855...> done.
 Note: <http://www.ebi.ac.uk/ontology-lookup/?termId=CHEBI:16855> returns formula C6H14N2O2

Figure 3.8: The database tab. This tab allows to add new models in the SBML format to the database. Furthermore, it allows to investigate the origin and interconnection of merged substance entries. In the image, D-Lysine and its referencing data sources are shown.

strings (c.f. 5.3.2) contained in the data sources. The results of this comparison are shown in a tree and visualized in the right part of the tab. The *Information* tab shows a few lines giving the author of the tool and that its results should be used with care.

3.4.4 The Server

The **server** part of the toolbox consists of a few classes that coordinate the execution of tasks. It therefore manages a list of registered client threads together with their current state. At startup it sets up a TCP server socket on which it listens for clients that want to connect to this server. The connected clients are then used to perform the calculation on submitted tasks. A **task** is a specific problem case equipped with problem specific parameters. The server takes the parameters from the GUI input fields and creates task objects according to the task button activated in the front end (Figure 3.4). These objects are then collected in a task queue and are sent in FIFO order to those clients that are marked as idle. For submitting the tasks to the clients, the task data is serialized. Each client that has received a task is then marked as busy until a result or abort-of-task signal is received from it. Results are received as serialized objects which are then unserialized and put in the result list provided for the GUI (Figure 3.6). A refresh event is sent to the graphical front end upon each reception of a result object.

3.4.5 Clients And Optimization Routines

This section describes the optimization routines that were planned to be usable with the help of this tool and that have been implemented as well as the problems that occurred. These optimization routines were designed to be executed by the clients. Therefore, they were prepared as tasks set up within the *Tasks* tab of the GUI (Figure 3.3). Pushing a task button in the front end triggers the submission of a serialized task to a client.

3.4.5.1 Search For Processing Organisms

This is a rather basic task and can be described as follows: The user selects a certain substance listed in the database via the “selectable substances” list (Figure 3.4) and adds it to one of the user lists. This list is then marked with the “substances to

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

degrade” option. After pressing the “search for processing organisms” button, a `ProcessorSearchTask` object is created and sent to a client. This client then searches the database for all reactions containing the selected substance as an educt (for reversible reactions, also the products are checked). For each reaction found, the catalyzing enzymes are searched and for each enzyme the compartments or organisms containing that enzyme are requested. The list of compartments found is then returned to the server and displayed in the “Results” tab of the GUI. This routine implements the first step within the degradation pathway search, described in Section 3.3.1.

3.4.5.2 Calculate Products

This is another very basic task based on a list of user selected species and user selected substances. For each species a separate task is generated which aims to calculate all substances that can be produced when the organism is supplied with the selected substances. This calculation is based on a graph reachability approach. This implies that the result is not based on a network path with balanced intermediate substances. The search works as follows: For each organism the complete list of reactions is read from the database. Then, starting from the available substances, for each reaction and each allowed direction the feasibility is checked by testing whether the reaction’s educts are contained within the available substance set. For each feasible reaction, the products are added to the list of available substances. These steps are iterated, until no further feasible reaction can be found. The list of available substances is then returned as the calculation result. As this list of products is not based on a balanced flow, it is no answer to the question of aggregating intermediates, but it may help to clarify if a certain substance can be within the product set.

3.4.5.3 Calculation Of Additional Maximizing The Set Of Products

Like the two routines described before, this is a network-based approach. It is based solely on the compartment list selected by the user and the substance list marked as “substances to degrade”. For each of the compartments the algorithm does the following:

1. The full set of reactions and the list of involved substances assigned with the organism is obtained from the database.

2. The set of substances reachable is calculated as described in Section 3.4.5.2.
3. For each substance involved in a reaction of the organism, but not yet reachable, the respective product set, gained when the substance is added, is calculated.
4. The cardinality of substances already reachable is compared with the cardinality of the set extended by the not-yet-reachable substance.

Then for each not-yet-reachable substance the increase in the product set is returned as numerical value. With this method, it is possible to find the “most useful” substances to be added to an organism’s nutrient set. This calculation is not a solution to any of the subproblems of the main degradation problem, but proves rather useful to identify substances useful for an organism to grow.

3.4.5.4 Calculate Seeds

This optimization tries to calculate a minimal set of substances which can be supplied to the network to generate all substances collected in the “substances to produce” list. The term “seed” is lent from Borenstein et al. (5) and describes “*the minimal subset of the occurring compounds that cannot be synthesized from other compounds in the network (and hence are exogenously acquired) and whose existence permits the production of all other compounds in the network*”. Although this routine tries to find a solution producing all possible compounds if the list of “substances to produce” is empty, our approach has a focus on producing a set of desired substances defined in this list. For each compartment in the list of user-selected species, this is essentially done by a modification of the integer linear programming (ILP) method searching for elementary flux modes proposed by de Figueiredo et al. (15). In our adaptation, the seed set is the set of consumed substances of a flux mode that produces the desired products while minimizing the inflows. This is reflected in additional constraints of the form that the production rate of the desired products is forced to be positive and in an additional weighted term for the minimization of all inflows. The according weights can be adjusted within the parameters form (Figure 3.5). Based on whether the “Use MILP” box is checked or not, the restriction to integer variables is applied or relaxed to use real variables. This optimization was implemented with the intention to solve the problem of predicting the nutrients that additionally need to be added to the soil to allow a particular organism to grow on the available medium.

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

3.4.5.5 Calculate Flow Distributions For Given Input/Output

This routine is in turn an adaptation of the Calculate Seeds method. It applies two further constraints to the linear program:

1. For all substances present in the user-list having the “substances to degrade” box checked, it forces the inflows to be positive.
2. It tries to maximize the outflow of all substances listed in the “substances to produce” list while at the same time trying to minimize all other flow rates.

Thus, it tries to find flux modes taking up the substances to be degraded while minimizing the effort to build all substances that shall be produced. Note that this is a multicriterion optimization, where the importance of minimizing all flows, the importance of minimizing the inflows and the importance of maximizing the desired outflows are weighted according to the values set within the parameters form (Figure 3.5). Again, the user has the option to choose between a linear program using real numbers and a program using integers. Settled with biomass compounds (21) as desired products and the list of “substances to be degraded” as educts, this method aims to predict which intermediates would aggregate when an organism is forced to eat off the given pollutants. The intention is to solve the problem described in Section 3.3.4.

3.4.5.6 Paths From “Substances To Degrade” To “Substances To Produce”

This routine was intended to find the shortest reaction path connecting all substances in the list marked with the “to degrade” checkmark with all substances in the “produce” list. For this purpose, a breadth-first search has to be done on the network of each compartment provided in the user-selected species list: For each substance in the list of substances to produce, reactions that consume the substance are gathered and the products of the respective reactions are added to the list of reachable substances. This is iterated until all substances that are to be produced are in the list of reachable substances, and all primary educts have a connection to at least one of the connected components of the found subnetworks. This routine is not directly related to any subproblem of the degradation challenge, but was conceived as a valuable add-on to the toolbox.

3.4.5.7 Calculate Additional With Evolutionary Algorithm

This button triggers the start of an evolutionary algorithm (9) for each compartment in the user-selected species list: Each algorithm instance takes the list of “substances to degrade” and the list of “substances to produce” and tries to find a set of substances supplied to its assigned compartment which contains all the desired inflows and produces all the desired outflows while minimizing the total number of inflows. Therefore, it mutates the set of supplied inflows by adding and removing substances by chance. The fitness function for the evolutionary algorithm is given by:

$$fitness = \frac{degradationSuccess \times productionSuccess}{sizeofactualinflowset} \quad (3.1)$$

$$degradationSuccess = \left(\frac{\text{number of desired inflows in actual inflow set}}{\text{number of desired inflows}} \right)^2 \quad (3.2)$$

$$productionSuccess = \left(\frac{\text{number of desired products reachable}}{\text{number of desired products}} \right)^2 \quad (3.3)$$

The evolution step is carried out for t timesteps, with t being

$$t = speciesCount \times \left\lceil \frac{10}{\ln(speciesCount)} \right\rceil,$$

where *speciesCount* is the number of species potentially involved in any reaction assigned with the respective organism. This measure has been chosen after trial-and-error in some test cases; here the procedure yielded acceptable results. This evolutionary algorithm was added to the toolbox as an alternative to the flux based approach: Testing the latter, we realized that linear optimization in some cases does not work. These failure cases are owed to invalid stoichiometric information in some datasets gathered from online databses.

3.4.6 Exemplary Workflow

For explanation purposes, a sample workflow will be described here. The following descriptions require that an instance of the Interaction Toolbox and at least one client has been started.

- Calculating the products potentially producible from a set of substances.
This task at first requires one or several organisms to be selected. This can be done in the “selectable species” tab within the “Tasks” tab. For the example

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

“E. coli iJR904” has been chosen. Clicking on this organism reveals further details, whereas double clicking (or pressing the “ \Rightarrow ” button) adds the species to the “user selection” list on the right hand side (Figure 3.3). Immediately after the organism has been put in the “user selection” list, all substances annotated to be part of the known metabolic network of this organism are provided in the list of selectable substances. From here, specific substances can be inspected by clicking or added to the active “User selection” list by double-clicking on entries. Figure 3.4 shows D-Glucose and three subordered compartment-specific synonyms with some details added to the list of “substances to degrade”. In the same way, other substances can be added to each of the user selected substances lists. After that, different actions tasks can be started by activating the respective button in the tasks column. When pushing the buttons “Calculate products” and “Calc additional maximizing the set of products”, these tasks are sent to the client(s), and after a while calculation results are received. This can be observed by the messages in the status area at the bottom of the user interface. The results are displayed in a tree-list within the “Results” tab (Figure 3.6) and can be explored using the mouse.

- Tracking the network connectivity of a chemical compound of interest.
Upon right-clicking on a substance in any of the lists, a context menu opens, giving the possibility to search Google for this substance or add the respective substance to any of the user selection lists. Furthermore, the menu contains an entry with the caption “show in network view”. Choosing this option loads the reactions producing and consuming the selected compound (Figure 3.7).
- Finding bacterias with a potential capability to degrade a substance of interest.
For this purpose, the “Tasks” tab and the “selectable substances” tab need to be opened. In the latter tab, at the bottom, next to the “clear list” button, there is a drop-down text field. Here a part of a name of a compound of interest, for example “Benzene”, can be entered. While this is entered, the database is crawled for substance names containing the given text, and the results are provided in the drop-down list. Upon selection, the substance is automatically added to the substances list above. Double-clicking on a substance in the left list adds it to the currently opened list on the right hand side. By default, the first list on the right

hand side has the “substances to degrade” checkmark set. Having this activated, the button “Search for processing organisms” can be pressed, and after a short calculation time, a list of organisms exhibiting reactions that act on benzene should appear in the list on the “Results” tab.

- Comparing substance entries from various databases.

As described before, the Database tab lists all compound entries of the local database that originate from multiple databases and have been merged. Clicking on each of them starts an algorithm analyzing the pristine source database entries and showing comparison results. For example, when D-Lysine is selected, all eight sources featuring this substance are compared against each others and a hierarchical list of similarities is prepared. Browsing this list by mouse reveals that the ChEBI and 3DMET databases offer different SMILES strings for this substance, which is, in turn, represented by a red line in the graph on the right hand side (Figure 3.8).

3.5 Calculation Problems

The greatest challenge in the preparation process of this toolbox was the design and filling of the database. Parsers for various page and file formats have been implemented to read fields like name, chemical structure, stoichiometric parameters, relations between entities within one database and links between several databases. The first difficulty encountered was the fact that some chemicals have divergent formulas in different data sources. This was overcome by interactive decisions by the user as described before. But even within a single database inconsistencies regarding molecular formulas were found. This especially led to reactions not balanced on an atomic level, which trace back to wrong or inappropriate formulas for some molecules. These inconsistencies did not interfere with the graph-based calculation approaches described in Section 3.4.5.1, 3.4.5.2, and 3.4.5.6, as those graph algorithms solely rely on the relations of molecules and reactions. Still, huge problems with the flux based linear optimization approaches were experienced, since these approaches are additionally based on molecule stoichiometry. Problems often occurred where the ILP solver reported unbounded or unfeasible problems working on the KEGG based models, while it worked satisfactory on our test cases. Finally, a model that was predicted to be able to create amino acids without

3. NETWORK DATA INTEGRATION AND ANALYSIS SOFTWARE

any source of nitrogen helped to identify the source of suchlike problems: KEGG contains several simplified reactions that coalesce chains of subsequent reactions. Many of them are marked as “unclear”, “unknown mechanism” or “multi-step reaction”, yet unfortunately this is not the case for all of these reactions. Eventually this motivated us to perform a more detailed analysis of the KEGG database and the inconsistencies therein, as described in the next chapter. Due to the problems described there, we were not able to conduct an operational check on the linear optimization routines for large scale models. Also, we were not able to perform these tests on curated networks, since database investigation was then prioritized and further development of the toolbox was suspended.

4

INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

4.1 Overview

As mentioned in the previous chapter, we need high accuracy data in order to perform automatic analyses and reliable model construction approaches. This chapter analyzes the inconsistencies and problems within online metabolic databases. It is motivated by the problems encountered while writing the toolbox described in Chapter 3. Thus, The Kyoto Encyclopedia of Genes and Genomes (KEGG) was chosen as a widely used database for which problems are examined and potential mitigation strategies are sketched out. In the first half of this chapter, we present our computational approach for classifying inconsistencies and provide an overview over detected inconsistency classes. In the succeeding part, we present strategies to deal with the problem classes detected. We identified inconsistencies both for database entries referring to substances and entries referring to reactions. We especially propose a rule-based database approach allowing for the inclusion of parametrized molecular species and parametrized reactions. This will be explained in more detail in the following chapters. Detailed case-studies for the alcohol dehydrogenase reaction and nucleic acid chemistry, respectively, are then used to demonstrate the applicability of the approach.

The contents of this chapter are contained in the paper “Towards rule-based metabolic databases - a requirement analysis based on KEGG”, which has been submitted to the International Journal of Data Mining and Bioinformatics.

4.2 Introduction

Chemical reactions are the backbone of almost any biological process, and to investigate their interrelationships has been of interest for understanding biochemical pathways in various organisms for centuries (7). In the last decades, huge amounts of data describing chemical reactions became available for research in online databases. Such data can be used for the reconstruction (12, 14) and exploration (26) of biochemical reaction networks as well as for the *de novo* construction of metabolic networks for particular applications (16). These networks can in turn be used for predictive tasks (5, 13), analyses using the petri net framework(47) or the investigation of metabolic interfaces (4) as well as a variety of other approaches(56, 62).

For tool chains which access databases to automatically assemble interaction networks, we need to rely on the quality of the respective data. In flux-based pathway analyses (FBA)(51) of pathways assembled from publicly available databases, physically impossible results may occur. For instance, some of the assembled pathways might be able to generate new elements because the associated reactions stored in the database are not elementally balanced, i.e. the abundance of elements on educt and product side differs. As much of the data contained in online databases has been acquired semi-automatically, a certain amount of inconsistencies can be expected, inevitably limiting the use of the database information for any automated pathway analysis.

The aim of this study is the assessment of such inconsistencies concerning the balance of elements. Besides identifying and classifying these, we also present approaches which might reduce the occurrence of such inconsistencies in the future. To introduce this concept of assessment, we are using KEGG as a representative database; still the concepts will also be applicable to other databases of similar scope and content ((40)). For KEGG, some quality-related issues have been analyzed before, for example annotational errors related to partial EC numbers (25) and problems with the structural representation of molecules(52). Also, problems with reaction balances have been examined before(22, 42), and some publications recognized that, next to simple problems with formulas missing one or two atoms, there are serious problems with polymer reactions and generic substances (22, 42, 52). While these previous studies, in particular a short review of reactions that have been identified to be unbalanced in the past (22), enable us to get an impression of the progress of error correction in KEGG, they did not

provide a solution strategy. We thus complement our analysis of problem classes with mitigation strategies and in particular with a formal rule-based scheme to circumvent current ambiguities.

4.3 Methods

4.3.1 Structure Of KEGG

KEGG is a comprehensive online database which integrates organismic, genetic, and molecular data with extensive information on enzymes, reactions, and metabolic pathways (39). After its foundation in 1995, it has been growing steadily and today serves as a powerful source of knowledge in OMICS research. All data sets belong to one of the three categories “Systems Information”, “Genomic Information”, and “Chemical Information”. The “Systems Information” pages provide integrated overviews on pathways, maps, drugs, and others. “Genomic Information” describes genes, genomes, and organisms, while “Chemical Information” subsumes data on general chemical compounds, glycans, ligands, and reactions. All information regarding chemical species are given in structured tables containing, besides other fields, rows for the entries “IDs”, “names”, “formulas”, “equations”, “structure”, “remarks”, and “comments”. Entries in all groups of data are cross-referenced. Although some of the data of KEGG are available only to paid subscribers, large parts are freely accessible via a web interface and a ReST API (39, 67).

4.3.2 Local Data Management

Our analyses are based on those parts of the KEGG database which are freely available. The data used for automatic analysis have been downloaded from KEGG with the help of a Java programme (**Database Filler**) that parses data retrieved from KEGG via the ReST API¹ and stores relevant information in a local MySQL database. A local database for analysis appears to be the best choice to speed up the analysis algorithm, as each online API page request can take up to one second.

The local database to store KEGG data is structured as shown in Figure 3.2 and described in detail in 3.4.2. Initially, the **Database Filler** reads the list of all chemical

¹<http://www.kegg.jp/kegg/docs/keggapi.html>

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

species contained in KEGG, including “Compounds”, “Glycans”, and “Drugs”. Data entries are downloaded and for each of them the formula is stored in the substances table of the local database as a string. Subsequently, the reactions are downloaded and linked to the substances via the substrate and product table. Our analysis is based on six database snapshots which have been collected in July 2011, February, May, July, December 2012, and May 2013.

All tools for the download of data and for inconsistency analyses are available from http://www.biosys.uni-jena.de/interaction_tools.html.

4.3.3 Algorithm

4.3.3.1 Substance Checking

As the compound, glycan, and drug information is read and stored in the substance collection of the local database, the first consistency check is performed by the **Database Filler**: many compounds contained in KEGG have synonymous entries in the glycan or drug collection and are connected by “Same as” references. In some instances, cross-referenced entries in different collections contain empirical formulas that differ, leading to conflicting descriptions of the same compound. To detect such inconsistencies, empirical formulas which are given as alphanumeric strings in the “Equation” field of the respective dataset are parsed and stored in the local database. For all cross-referenced entries of compounds, drugs, and glycans, the respective formulas are converted to element-to-coefficient mappings; meaning that for each compound, drug, and glycan entry a table is created, mapping the occurring elements to their respective count. These mappings are then compared for each entry to identify compounds with non-unique formulas.

4.3.3.2 Reaction Checking

The analysis algorithm iterates through all retrieved KEGG reactions stored in the local database and calculates balances for each according to the respective reactants and products. Some reactions use substances for which no empirical formulas are available. We will refer to them as “indistinct” reactions. For the remaining reactions, the algorithm computes the atom counts based on participating substances for both the left and

4.4 Analysis Of Database Inconsistencies

date	reactions	substances
May 2013	9111	27,491
December 2012	8,958	27,526
July 2012	8,823	27,401
May 2012	8,748	27,110
February 2012	8,663	27,075
July 2011	(no data)	26,747

Table 4.1: Number of Extracted Entries of the KEGG Database at different times

right hand side; taking into account compound formulas and stoichiometric coefficients. For some reactions, polymerization reactions being a prominent example, stoichiometric factors contain variables such as n , $n + 1$, $n - 1$, or similar terms. To evaluate these reactions, however, these variable terms need to be resolved. In an initial scan of the whole database for subtrahends in the stoichiometry of the equations, no subtrahend larger than four was found. Thus, we chose to replace n by 5 during the balance checking procedure to resolve those variable terms in the reaction stoichiometries. To sum up the aforementioned points, the balance checking algorithm uses replacements like the following: n is replaced by 5, $n + 1$ becomes 5+1, which is resolved to 6, and $n - 1$ becomes 5-1, which is resolved to 4 (and $n - 4$ becomes 1). Now we distinguished the *balanced* reactions — for which the sums of the substrate atoms match the sums of the respective product atoms — from the *unbalanced* ones. An additional refinement extended the algorithm to divide the *unbalanced* reactions into those which are just (stoichiometrically) unbalanced and those which are “transmutational”: the first class held reactions for which the same *set of elements* appeared on both sides with different stoichiometries. The latter class collects cases of reactions where the substrate set also contains different elements than the product set.

4.4 Analysis Of Database Inconsistencies

4.4.1 Database Evolution

In May 2013 we extracted a total of 27,491 substances (compounds, glycans, and drugs) and 9111 reactions from the KEGG database. In the following this will serve as the

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

reference snapshot of the database when citing numbers. An overview over previous database sizes is given in Table 4.1. Note that the number of substances has decreased since December 2012, which has not been caused by a loss of compounds, glycans or drugs, but by an increased number of unifications made possible by various corrections of formulas in KEGG.

4.4.2 Inconsistent Compound References

The Data In the data set acquired on May 31, 2013, we found 12 synonymous, cross-referenced (“Same as“ link) sets of compounds, drugs, and glycans that had differing formulas. In many cases, this was caused by the duplication or polymerization of basic structures as it appears with KEGG compound C00718 and KEGG drug D02329. Deviations might also be caused by associated ions that are annotated in only one element of the set. Those sets are listed in Supplement 8.1, Table 8.1.

Proposed Solution Strategy A short term solution is to replace all “Same as” links between compounds, glycans, and drugs with different formulas automatically by a new link type “Related to”. On the long run, adding a dedicated field for an “Is a”-relation will allow for the handling of specific and more generic chemical species in a coherent way. After this has been implemented, it will, for example be possible for a reaction to use a generic compound “Primary Alcohol” (C00226), while various alcohols like methanol or ethanol refer to this generic compound via the “Is a”-relation. Additionally, for the class-representative compounds there ought to be back-references to the particular substances in the database. A lot of the compounds (and also reactions, enzymes, and other entries) in KEGG are already integrated within KEGG BRITE (39) which aims to collect functional hierarchies of metabolic entities. Unfortunately, the position of an entry within the BRITE database is not reflected on the entry pages until now.. Such an annotation would casually extend the database towards an ontology like the ones implemented in ChEBI (28) or Rhea (1) — which, in turn, is considered to be beneficial (2) as it enables reusable integration of knowledge and mediation between different platforms. We propose a residue database as explained in more detail with the example of generic and concrete alcohols in Section Section 4.5.1.

4.4.3 Balanced Reactions

7087 reactions (77.79%) were found to be well balanced in May 2013. For these reactions, a chemical formula is given for each reactant where the number of atoms on the right hand side of the equation matches the number of respective atoms on the left hand side of the equation.

4.4.4 Indistinct Reactions

The Data In 723 reactions (7.94%) we found at least one agent for which no chemical formula was available from the KEGG database; meaning that the reaction balances could not be tested. This is frequently the case for generic compounds, which are used as placeholders for a whole set of substances; for example KEGG compound C00030 which is a “reduced acceptor / hydrogen-donor”. Also, substances that share a certain functional group are often subsumed under a single compound, like “ferredoxine” or “thiol”, and have no common formula. The list of indistinct reactions is contained in Supplement 8.2, Table 8.2.

Proposed Solution Strategy The “Is a”-relation recommended in the above section on inconsistent compounds can also be used to interconnect substances, residues, and classes of residues to superordinate classes of residues. This, in turn, can be referenced by reactions, at least giving some more information and pushing the database towards an ontology. For some compounds — for which no formula is currently available — it is possible to propose empirical formulas if all other substances in related reactions are known and their empirical formulas are valid. To enable the software-based detection of compounds encompassing such auto-derived formulas, there should at least be a hint in the “comments” section or in a field meant exactly for this purpose. For those substances of which the formula is neither known nor derivable, the user will have to find a case specific solution strategy depending on the respective application.

4.4.5 Unbalanced Reactions

1114 reactions (12.22%) have been found to be unbalanced. The full list of unbalanced reactions can be found in Supplement 8.3, Table 8.3. These imbalances can be

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

assigned to four problem classes: *Inappropriate Annotation*, *Polymer Reactions Containing Quantifiers such as n or m* , *Disappearing Protons*, and *Other Reactions with Mismatching Atom Counts*.

4.4.5.1 Inappropriate Annotation

The Data Some generic compounds are depicted as a functional group attached to a pendant group symbolized by the pseudo “element” **R**. While the real formula differs for all of these compounds or may even be unknown, they can be subsumed to a class with a common formula using this residue notation. In comparison to not giving a formula at all and thus being left without any possibility for stoichiometric testing, this is indeed an improvement. However, for many cases this is still insufficient:

if, for example, two substances are to be examined which participate in one reaction and are stored this way, but with different “fixed” molecule parts symbolized by the same symbol **R**, it is obvious that the balance cannot be checked. An example for an unbalanced reaction due to **R** referring to different entities is the *Acyl-CoA: oxygen 2-oxidoreductase* reaction (KEGG R00388, Figure 4.1a), where acyl is represented by **R** on the left hand side, whereas trans-2,3-dehydroacyl is represented by **R** on the right hand side.

Proposed Solution Strategy The “Is a” relation suggested above also proves beneficial here. Additionally, for residues that are the same in succeeding reactions but divergent in different reaction pathways, we suggest to create a residue collection, making it possible to distinguish residues even without assigning an explicit formula. This is, for example, important for methods like, for example, flux balance analysis: in these cases, one usually cannot simply use a common reaction and common substances here. For FBA models it is necessary to include and distinguish all possible instances of a reaction class and build separate mass balances for all respective reactants.

4.4.5.2 Polymer Reactions Containing Quantifiers Such As n Or m

The Data This subclass of unbalanced reactions is formed by the elongation or a shortening of polymers: a polymeric substance is annotated on both sides of the reaction and usually one substance acting as a building block for the polymer is added on one side of the reaction equation. Therefore, there is a need for a variable which

4.4 Analysis Of Database Inconsistencies

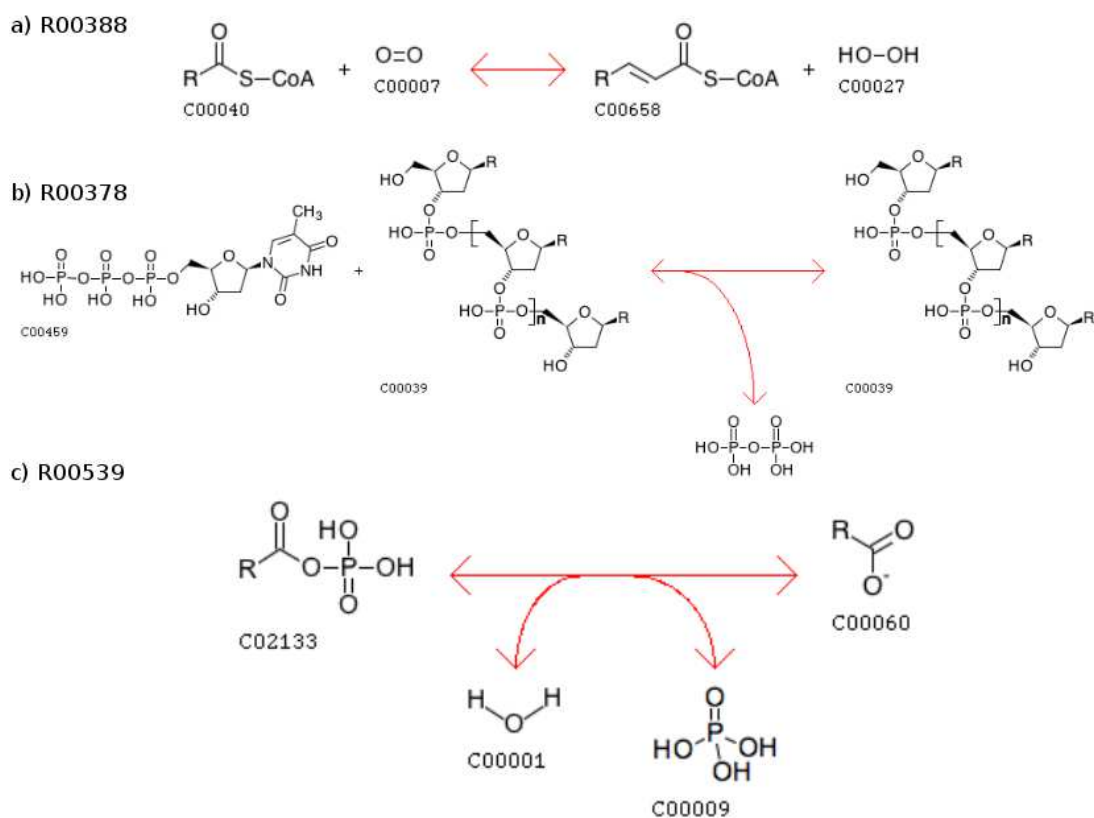


Figure 4.1: Examples of Classified Unbalanced Reactions from the KEGG Database: a) reaction with molecules holding different residues: acyl is represented by **R** on the left hand side of the *Acyl-CoA:oxygen 2-oxidoreductase* reaction (KEGG R00388), while trans-2,3-dehydroacyl is represented by **R** on the right hand side
 b) the *deoxythymidine triphosphate:DNA deoxynucleotidyltransferase* reaction (KEGG R00378) is an example of an improperly modeled polymer reaction. Note that it should be n+1 on the right hand side.
 c) loss of proton in the *acyl phosphate phosphohydrolase* reaction (KEGG R00539)

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

represents the polymer length and differentiates between both sides. However, quite often this is not taken into account, as in the *deoxythymidine triphosphate:DNA deoxynucleotidyltransferase* reaction (KEGG R00378, Figure 4.1b).

Proposed Solution Strategy There already is an, albeit very limited, approach to circumvent this problem: some of the polymer reactions are already parametrized, for example the *Deoxynucleoside triphosphate:DNA deoxynucleotidyltransferase* reaction (KEGG R00379). Yet, here the chain length parameters are only added to the name fields of the respective substances, an approach which is of limited use in other contexts and does not provide any further information about n . Further, there is no integration of parameter dependency in the BRITE hierarchy. At the moment there is no clear formalism describing how such parameters are to be applied. In general, such parameters should be declared independently of substance names and used more consequently. Furthermore, a common formalism for multi-parametrized formulas is to be constituted, as sketched out in Section Section 4.5. Unfortunately, within the reactions already parametrized, we can find cases where the stoichiometry is only valid for a certain value of n , even though it should be valid for arbitrary values. In some cases the reaction is not balanced for any value of n . Due to these problems it seems advisable to implement an automated balance check.

4.4.5.3 Disappearing Protons

The Data The third class are reactions where a single proton or a pair of protons is omitted due to the formation of an ion, reflected by a missing **H**, **H**⁺ or comparable occurrences in the formula, as in the *Acyl phosphate phosphohydrolase* reaction (KEGG R00539, Figure 4.1c).

Proposed Solution Strategy These occurrences are often due to the protonation or deprotonation of conjugate acid-base pairs. Depending on the pH value, a substance can exist in several charge and protonation states. Nonetheless, this pH dependency is not captured in the database, and up to this point there is no standard approach on how to deal with this problem. Also, the pH value dependency of protonation states is just one symptom of another, even more common problem: depending on environmental settings, many substances may change their charge, reactivity, and conformation. This

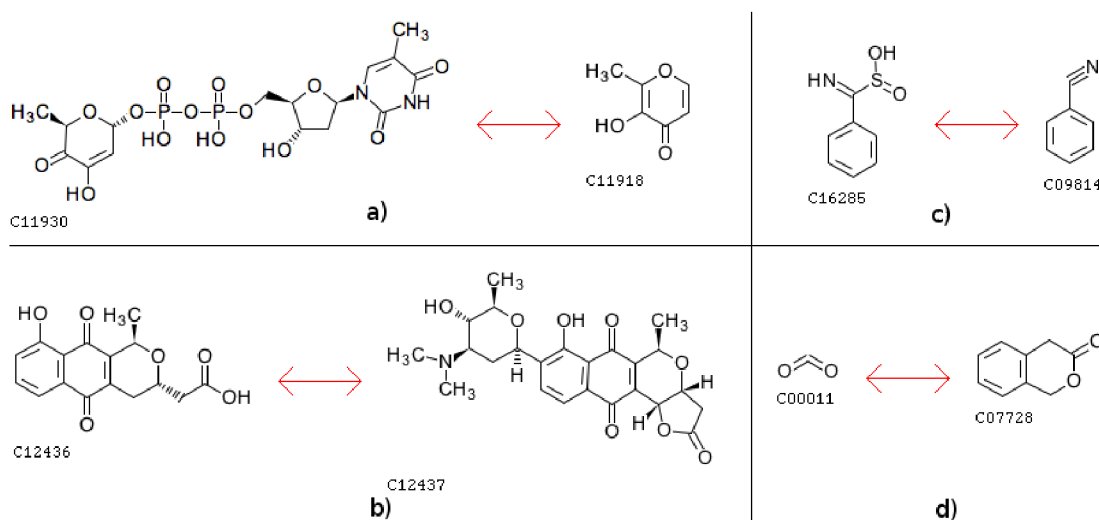


Figure 4.2: Examples of Unbalanced Reactions not Belonging to any of the Aforementioned Classes (cp. Fig. 4.1): a) the *untitled* reaction (KEGG R06443), where a larger molecule “collapses” into a single ring, b) the *untitled* reaction (KEGG R06698), where a large part of the structure just disappears, c) the *untitled* reaction (KEGG R07838), where a side-chain is completely replaced, d) the *untitled* reaction (KEGG R05539), showing the conversion of a small structure to a double ring.

should be recorded in the database, either by defining standard conditions or by giving the reasonable ranges of conditions. In any case, if a substance is stored in the database with its empirical formula, this formula refers to a particular fixed state which should, in turn, be reflected in each occurrence of that substance in a reaction entry. A lack of single protons can easily be detected, and in most cases an automated correction of these reactions is possible; see also Ott and Vriend (52). In practice, these automatically corrected reactions should then be tagged until they are manually validated.

4.4.5.4 Other Reactions With Mismatching Atom Counts

The Data The last class contains the set of reactions with other errors causing inconsistencies in the formulas (see Figure 4.2 for examples). Often, this is the case with incomplete or multi-step reactions for which, up to this point, the exact mechanism is unknown. In most — but not all — cases, additional information about this lack of knowledge is given in the reaction’s commentary field. Unfortunately, as this is text

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

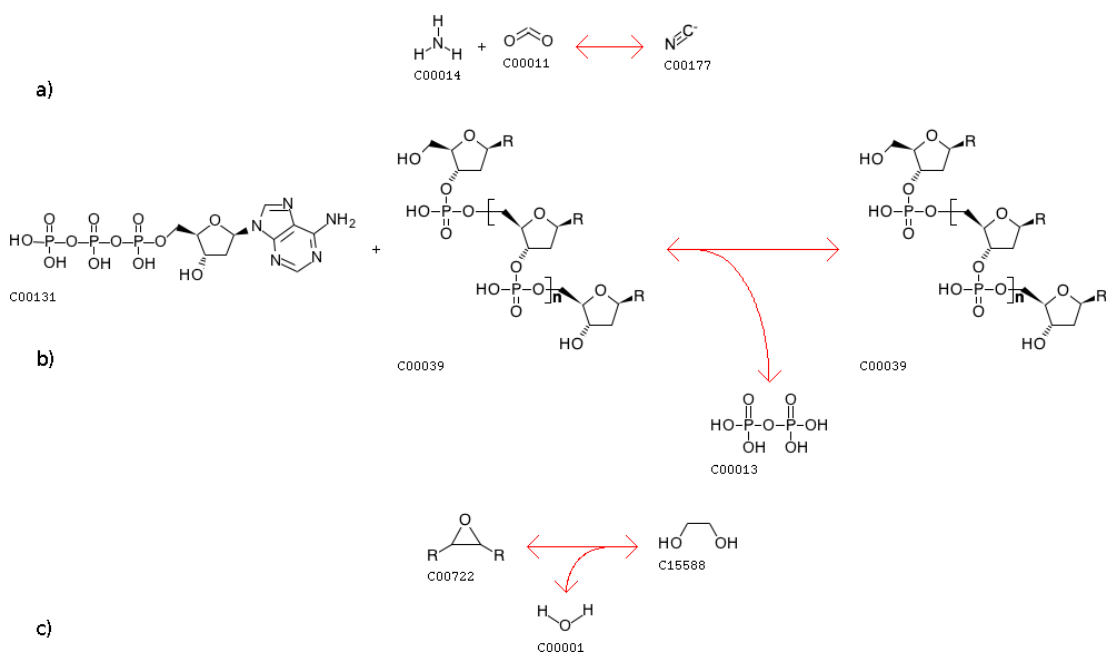


Figure 4.3: Examples of transmutational reactions:

a) the *untitled* reaction (KEGG R00152): hydrogen and oxygen dropped

b) the *Deoxyadenosine 5'-triphosphate:DNA deoxynucleotidyltransferase* reaction (KEGG R00375):

adenine converted to residue

c) the *epoxide hydrolase* reaction (KEGG R02822): loss of residue

readable only by humans, an automatic evaluation of a reaction's validity in these cases becomes difficult.

Proposed Solution Strategy As reactions of this class are inconsistent due to a variety of reasons, there is no possibility to automatically correct them. However, it is possible to calculate the deviance between the right hand side and the left hand side of the formula. This deviance can then be displayed next to a raised “unbalanced” flag.

4.4.5.5 Transmutations As A Peculiar Subclass Of Unbalanced Reactions

Obviously inconsistent are those reactions in which there are differences in the sets of elements that appear on the substrate and product side. Thus one can find elements on the left hand side which are not present on the right hand side of the reaction or vice versa. We refer to this subclass of unbalanced reactions as “transmutational reactions”

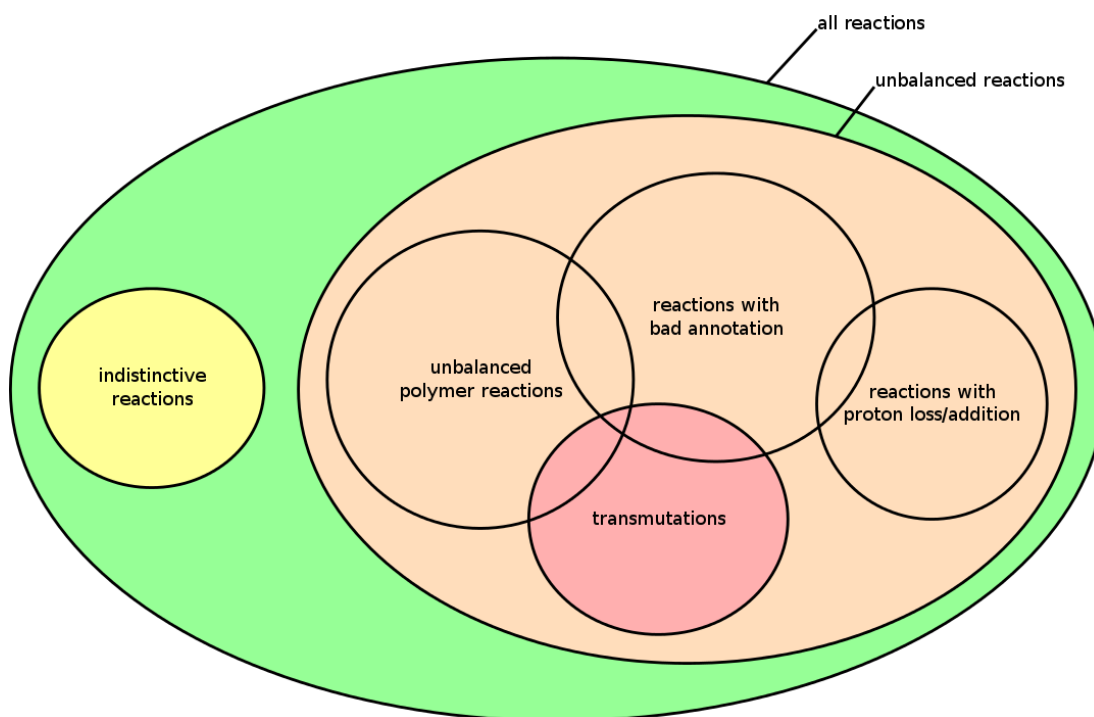


Figure 4.4: Reaction Classes and their Major Subsets - Not to Scale

Note that only the green area denotes balanced reactions, i.e. all reactions without indistinct and unbalanced reactions. This venn diagram illustrates the class composition and relations; the size of the distinct areas does not relate to the number of entries within.

as they seem to convert elements — a process known as transmutation (31). Our algorithm detected 193 of these reactions in the KEGG database. Again, most of them are just incomplete or multi-step reactions with yet unknown mechanisms, for example KEGG reaction R00152 (Figure 4.3a), where oxygen and hydrogen just disappear. In other reactions, again, there seem to be annotational problems as atoms or groups are casted into an **R** “element”, for example in the *Deoxyadenosine 5'-triphosphate:DNA deoxynucleotidyltransferase* reaction (KEGG R00375, Figure 4.3b). In some rare instances, remaining molecule parts depicted by an **R** symbol just disappear, as in the *epoxide hydrolase* reaction (KEGG R02822, Figure 4.3c). Again, most of those reactions are tagged for incompleteness in the “comments” section, but in some instances such information is missing. Here, the solution strategies proposed in the previous sections on unbalanced reactants also apply.

A condensed overview on the reaction classes and their relations is shown in Fig-

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

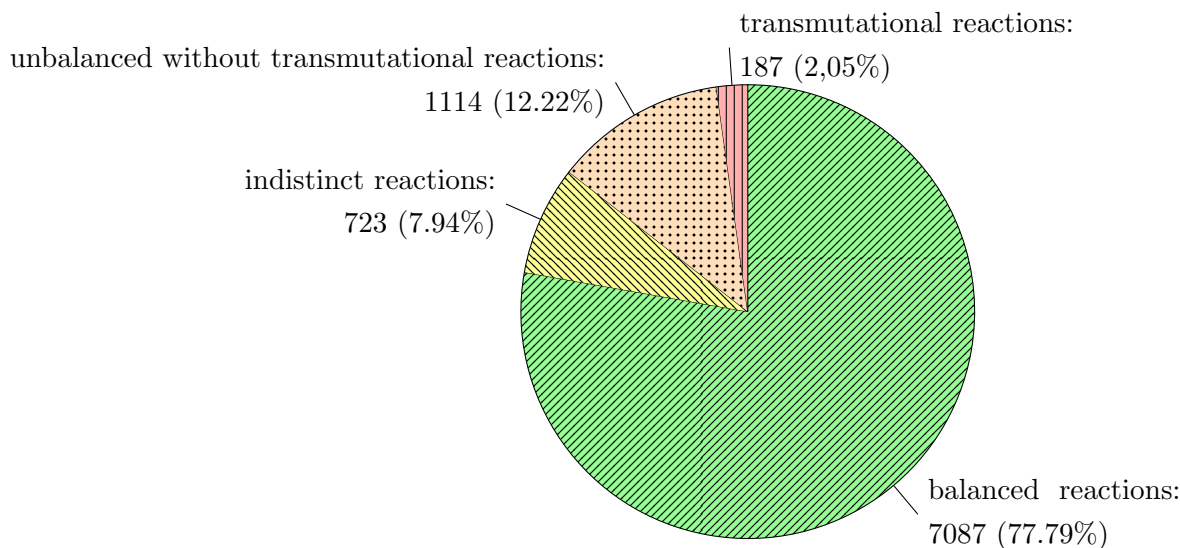


Figure 4.5: Classification of KEGG Reactions, as Retrieved on May 2013.

ure 4.4. A chart showing the percentages is given in Figure 4.5. Proposed solution strategies are summarized in Table 4.2.

4.4.6 Reviewing Previously Identified Database Inconsistencies

Database entries are continuously undergoing curation. Here, we review some of the reactions which have been identified as being incorrect in previous analyses. For example, the *cellulose cleavage* reaction (KEGG R02886) was considered incorrect (22). The authors pointed out that the empirical formula for cellulose was wrong and that the reaction equation was incorrect. Since then, the empirical formula has been corrected, yet not the reaction equation. Another substance, KEGG compound C00369, commonly known as *starch*, was revealed wrong in (22) and has been corrected in the meantime. Similarly, the *alpha-D-Glucose-1-phosphate:alpha-D-glucose-1-phosphate 4-alpha-D-glucosyltransferase* reaction (KEGG R00957) has been corrected by adding water to the substrate list. However, a number of 16 reactions considered faulty in (22) still remain, namely R03873, R01762, R01790, R02184, R02185, R02186, R02187, R02188, R02189, R02421, R02886, R02887, R02888, R02889, R02890, R06046.

4.5 Towards Rule-Based Metabolic Databases

case	example	strategy
balanced	-	lean back and relax
indistinct	R15001	automated derivation + tagging, residue/substituents collection, ontology
unbalanced, incl. transmutational	bad annotation	R00388 residue collection, parametrized annotation syntax
	polymerization	R00379 unique, parametrized annotation syntax
	proton exchange	R00539: $CH_2O_5PR + H_2O \rightarrow H_3PO_4 + CO_2R^-$ automated correction + tagging
	others	automatic detection + calculation of difference + tagging

Table 4.2: Inconsistent Reactions and Corresponding Strategies

4.5 Towards Rule-Based Metabolic Databases

In this section, we first show how a formerly mentioned residue database could be used in an actual example case. Afterwards, we extend this example using a rule based formalism and explain it using another exemplary set of compounds, substituents and reactions. Using the following approaches, it is possible to model reactions in a consistent way, even if they operate on substances which are polymers or contain exchangeable residues.


4.5.1 Proposal For A Residue Database

Many reactions and chemical species possess common groups which are used in generic descriptions of species and reactions. For a coherent handling and reusability, we suggest a residue database providing residues together with their ID and formula. Such a residue database may be integrated in existing databases. In the following, we demonstrate the design as it could be integrated in KEGG.


4.5.1.1 Compound And Residue Set


As the ID scheme Rxxxxx is already used for reactions, a scheme of the form Sxxxxx is used here, referring to residues as **S**ubstituents. Using the residue database, a generic species entry can specify one or more sets of residues for the deviation of actual molecule instances. Here, we describe the example of the alcohol dehydrogenase reaction, and

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

Entry	C00226
Parameters	R
Name	primary Alcohol
Formula	CH ₃ OR
Structure	
Same as	C00132, if R=S10001 C00469, if R=S10002 C00756, if R=S10003 C06611, if R=S10004 C05576, if R=S10005
Comment	Type: generic compound in reaction hierarchy Generic alcohol, G can be any group
Reaction	R00623 ...
Enzyme	...
KCF data	show

generalization ↑
instantiation ↓

Entry	C00469
Name	Ethanol; Ethyl alcohol; Methylcarbinol
Formula	C ₂ H ₆ O
Exact mass	46.0419
Mol weight	46.0684
Structure	
Same as	D00068 D02798 D04855 D06542
is a	C00226[S10002]
Comment	IARC Group 1
Reaction	R00746 R00754 R02359 R02682 R04410 R05198 R09127 R09479 R09552
URNs	urn:miriam:cas:64-17-5 urn:miriam:pubchem.substance:3752

Entry	C00071
Parameters	R
Name	Aldehyde; RCHO
Formula	C ₂ H ₃ OR
Structure	
Same as	C00067, if R=S10001 C00084, if R=S10002 C01545, if R=S10003 C06613, if R=S10004 C05577, if R=S10005
Comment	Type: generic compound in reaction hierarchy Generic aldehyde, R can be any group
Reaction	R00623 ...
Enzyme	...
KCF data	show

generalization ↑
instantiation ↓

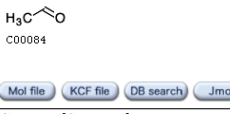
Entry	C00084
Name	Acetaldehyde; Ethanal
Formula	C ₂ H ₄ O
Exact mass	44.0262
Mol weight	44.0526
Structure	
is a	C00071[S10002]
Reaction	R00025 R00224 R00326
URNs	urn:miriam:cas:75-07-0 urn:miriam:pubchem.substance:3384

Figure 4.6: Application of the Residue Database for Specifying Generic Species C00226 and C00071 (top) and Particular Instances C00469 and C00084 (bottom). Note that S10001 to S10005 refer to the residue database (Table 4.3). The entries are formatted in a KEGG-like manner to allow for an easy comparison with the respective KEGG entries and are used by the example given in Figure 4.9.

4.5 Towards Rule-Based Metabolic Databases

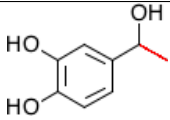
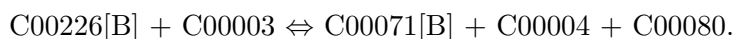
Residue ID	Comment	Structure
S10001	hydrogen	$H-$
S10002	methyl group	H_3C-
S10003		$H_3C - (CH_2)_6-$
S10004		$Cl - (CH)_2-$
S10005		

Table 4.3: Example of a Residue Database Containing Five Residues used in the Alcohol Dehydrogenase Reaction.

how generic alcohols can be stored; allowing for the consistent derivation of precise empirical formulas. In our example, the residue database consists of five groups with residue IDs S10001 to S10005 (Table 4.3). These residues can be parts of generic species, such as aldehyds and alcohols (Figure 4.6, top) and can be used as parameters to instantiate specific species such as acetaldehyde and ethanol (Figure 4.6, bottom). Additionally, we suggest the introduction of a “URNs” field for referring to external database entries to complement the existing “Other DBs” field.

4.5.1.2 Reactions Using General Molecules And Referencing Instances

For the alcohol example, we reformulate the the *primary alcohol:NAD⁺ oxidoreductase* reaction (KEGG R00623) to



Here, the parameter B represents the body of the alcohol molecule which may stand for one item of a limited set of possible residues.

Obviously, it is possible to use this format to formulate a reaction, both in a general and in a specific way; either by subsuming large sets of species in a superspecies used by general reactions or by the instantiation of free parameters with certain residues, respectively, yielding precise formulas for species and reactions. A more complete representation based on the KEGG entry of the reaction is shown in Figure 4.7.

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

Entry	R00623
Parameter	B
Name	primary alcohol:NAD+ oxidoreductase
Definition	Primary alcohol + NAD+ \Leftrightarrow Aldehyde + NADH + H+
Equation	C00226[B] + C00003 \Leftrightarrow C00071[B] + C00004 + C00080
Comment	general reaction NADP+ (ec 1.1.1.71, see R00625)
Same as	R00754, if B=S10001 R02878, if B=S10002 R05233, if B=S10003 R04880, if B=S10004 ...
RPair	RP00002 C00003_C00004 cofac RP00139 C00071_C00226 main
Enzyme	1.1.1.1 1.1.1.71
Pathway	rn00071 Fatty acid metabolism
...	...

Figure 4.7: Reshaped Entry for Alcohol Dehydrogenase Reaction

4.5.2 Introducing Rules

While the aforementioned changes would approximate some of the ontological functionality that is already present in other databases like Rhea(1), the following approach would further extend the capabilities by defining and applying rules: assignments of parameters to substances and reactions and the definition of ranges for these parameters as well as relations between associated parameters of interacting substances. To have a vivid impression of this framework, we use compounds and reactions related to nucleic acids and their elongation. As the following examples are intended to demonstrate the concept, biological details are omitted.

4.5.2.1 Compounds

We begin by defining a set of specific nucleobases: the compounds are *Adenine* (C00147), *Cytosine* (C00380), *Guanine* (C00242), and *Thymine* (C00178, Figure 4.8). These compounds will be used in the rule-based specification of nucleic acid molecules.

Substituents The substituent collection for this example will be rule-based because its entries represent molecule classes that are generic and can be used in various contexts. Again, we display a KEGG oriented structure and chose Sxxxxxx for the name spaces of the substituents. The most important novelty is the “Parameters” field and the rule-based specification in the “Same as” field; not only allowing for a specific usage in reactions but also for a generic notation.

Generic nucleobase Figure 4.9 shows the most general representation of a nucleobase (S00001). This entry may be used in all cases where a nucleobase is given without discriminating it further. It possesses one parameter c for instantiating the generic nucleobase by setting the parameter c to A , C , G , T , or U . With the help of the “Same as” field, the generic nucleobase is linked to various other entries: first, it is linked to the same general *nucleobase* entry (C00701) already present in KEGG; which, however, is not parametrized. Second, depending on the instantiation of the parameter c , we relate to the narrower substituents “pyrimidine base” ($c \in \{C, T, U\}$) and “purin base” ($c \in \{A, G\}$) and also to the concrete entries Adenine, Guanine, Cytosine, Thymine, and Uracil. For example, S00001[A] is the same as *Adenine* (C00147).

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

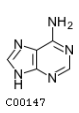
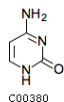
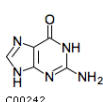
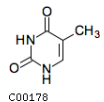
Entry	C00147 Compound	Entry	C00380 Compound
Name	Adenine; 6-Aminopurine	Name	Cytosine
Formula	C ₅ H ₅ N ₅	Formula	C ₄ H ₅ N ₃ O
Exact mass	135.0545	Exact mass	111.0433
Mol weight	135.1267	Mol weight	111.102
Structure	 C00147 Mol file KCF file DB search Jmol KegDraw	 C00380 Mol file KCF file DB search Jmol KegDraw	
Same as	D00034	Reaction	R00510 R00974 R02137 R02296
Reaction	R00182 R00190 R00194 R01244 R01245 R01401 R01402 R01561 R02557 R05708 R09675 R10185
...	...	Brite	Compounds with biological roles [BR:br08001] Nucleic acids Bases Pyrimidines C00380 Cytosine (Cyt) BRITE hierarchy
Brite	Compounds with biological roles [BR:br08001] Nucleic acids Bases Purines C00147 Adenine (Ade) BRITE hierarchy	URNs	urn:miriam:cas:71-30-7 urn:miriam:pubchem.substance:3670 urn:miriam:obo.chebi:16040 urn:miriam:knapsack:C00001498 urn:miriam:pdb-ccd:CYT urn:miriam:3dmet:B00097 urn:miriam:jscd:J9.343B
URNs	urn:miriam:cas:73-24-5 urn:miriam:pubchem.substance:3447 urn:miriam:obo.chebi:16708 urn:miriam:knapsack:C00001490 urn:miriam:pdb-ccd:ADE urn:miriam:3dmet:B00041 urn:miriam:jscd:J5.257D	KCF data	show
KCF data	show		
Entry	C00242 Compound	Entry	C00178 Compound
Name	Guanine; 2-Amino-6-hydroxypurine	Name	Thymine; 5-Methyluracil
Formula	C ₅ H ₅ N ₅ O	Formula	C ₅ H ₆ N ₂ O ₂
Exact mass	151.0494	Exact mass	126.0429
Mol weight	151.1261	Mol weight	126.1133
Structure	 C00242 Mol file KCF file DB search Jmol KegDraw	 C00178 Mol file KCF file DB search Jmol KegDraw	
Reaction	R01229 R01676 R01677 R01969 R02147 R03789 R10209	Reaction	R01411 R01412 R01413 R01414 R01415 R01570 R02806 R09937
...
Brite	Compounds with biological roles [BR:br08001] Nucleic acids Bases Purines C00242 Guanine (Gua) BRITE hierarchy	Brite	Compounds with biological roles [BR:br08001] Nucleic acids Bases Pyrimidines C00178 Thymine (Thy) BRITE hierarchy
URNs	urn:miriam:cas:73-40-5 urn:miriam:pubchem.substance:3541 urn:miriam:obo.chebi:16235 urn:miriam:knapsack:C00001501 urn:miriam:pdb-ccd:GUN urn:miriam:3dmet:B00067 urn:miriam:jscd:J9.344K	URNs	urn:miriam:cas:65-71-4 urn:miriam:pubchem.substance:3478 urn:miriam:obo.chebi:17821 urn:miriam:knapsack:C00001511 urn:miriam:pdb-ccd:TDR urn:miriam:3dmet:B00051 urn:miriam:jscd:J2.357D
KCF data	show	KCF data	show

Figure 4.8: Concrete Nucleobase Compounds used for our Rule-based Database Example. Adenine, Cytosine, Guanine, and Thymine entries are formatted according to the KEGG database, except for the URNs field. The changes apply to Uracil (C00106) in the same fashion, which is omitted here due to space restrictions.

4.5 Towards Rule-Based Metabolic Databases

Entry	S00001
Parameters	$c \in \{A, C, G, T, U\}$
Name	Nucleobase
Same as	C00701 S00002, if $c \in \{C, T, U\}$ S00003, if $c \in \{A, G\}$ C00147, if $c = A$ C00380, if $c = C$ C00242, if $c = G$ C00178, if $c = T$ C00106, if $c = U$
Reaction	...
Comment	represents one of the five nucleobases
Pathway	...
Enzyme	...
Brite	...
URNs	...
KCF data	show

Figure 4.9: A Rule-based Specification of a (Generic) Nucleobase as Part of the Substituent Collection. The “Same as” section links to a given nucleobase entry in KEGG. Note the parameter-dependent assignment to the particular nucleobases.

Entry	S00002
Parameters	$c \in \{C, T, U\}$
Name	pyrimidin base
Same as	C00380, if $c = C$ C00178, if $c = T$ C00106, if $c = U$
is a	S00001[c]
related to	C00396
Reaction	...
Comment	represents one of the three pyrimidin bases
Pathway	...
Enzyme	...
Brite	...
URNs	...
KCF data	show

Entry	S00003
Parameters	$c \in \{A, G\}$
Name	purin base
Same as	C00147, if $c = A$ C00242, if $c = G$
is a	S00001[c]
related to	C15587
Reaction	...
Comment	represents one of the two purin bases
Pathway	...
Enzyme	...
Brite	...
URNs	...
KCF data	show

Figure 4.10: Substituent Entries for the Generic Substituents “Pyrimidin base” and “Purin base”. They refer to their specific counterpart in the KEGG collection not by the “Same as” field but by “Related to”, because the KEGG entry uses a specific formula which does not match any instantiation of the respective generic classes (see also KEGG entry C00396).

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

Purin and pyrimidin bases Purines and pyrimidines are more specific classes of nucleobases. Still, they also exhibit a general facet, meaning that, for example, a *Purin Base* can represent Cytosine, Thymine, and Uracil. Therefore, we make use of a parameter c that may be any of the single letter abbreviations for these molecules (Figure 4.10). There already is a *pyrimidin base* compound C00396 and a *purin base* compound C15587 in the KEGG database. As their formulas are fixed and not taking into account the different incarnations, these are instances where we cannot use the “Same as” field but have to make use of the “Related to” field suggested above. A preview of those entries can be seen in Figure 4.10.

Hierarchy: From Nucleosides to Nucleic Acids Nucleic acid metabolism is both modular and hierarchical. Thus we are able to discriminate between further useful classes of components. By linking a sugar and a nucleobase we get a nucleoside, which, by means of combinatorics, may constitute one of ten specific molecules. These molecules can be grouped by parametrising the nucleoside as follows:

- parameter o determines the oxidation/deoxidation level (zero for DNA, one for RNA).
- parameter c determines which base is used (refer to section “Nucleobases”).

In Figure 4.11, left, we show two equivalent notational forms, as there might be different options of parameter annotation. Because we cannot give a common formula, we use the “Composition” field which allows for the exact calculation of an empirical formula by the instantiation of the parameters. Again, we relate the substituent to its corresponding *nucleoside* compound (C00801).

The next step toward a nucleic acid are nucleotides (Figure 4.11, right). These are basically phosphorylated nucleosides, where the level of phosphorylation is indicated by an additional parameter $p \in \{0, 1, 2, 3\}$. Once again we use the composition field to allow for the calculation of an empirical formula. By concatenating various phosphorylated nucleotides we get a *nucleic acid* (S00006, Figure 4.12).

The nucleic acid possesses only two parameters (Figure 4.12):

1. the phosphorylation level o , discriminating between DNA and RNA, and
2. the string S , representing the base sequence of the molecule.

4.5 Towards Rule-Based Metabolic Databases

Entry	S00004	Entry	S00005
Parameters (a)	$o \in \{0, 1\}$ $c \in \{A, C, G, T, U\}$ $(o, c) \notin \{(0, T), (1, U)\}$	Parameters	$(o, c) \in \{(0, A), (0, C), (0, G), (0, U), (1, A), (1, C), (1, G), (1, T)\}$ $p \in \{0, 1, 2, 3\}$
Parameters (b)	$(o, c) \in \{(0, A), (0, C), (0, G), (0, U), (1, A), (1, C), (1, G), (1, T)\}$	Name	(deoxy)Nucleotide
Name	(deoxy)Nucleoside	Composition	S00004[o,c] (HPO3)p
Composition	C5H8O(3+o) S00001[c]	Reaction	...
Reaction	...	Comment	either phosphorylated deoxynucleoside (o=0) or phosphorylated nucleoside (o=1)
Comment	either deoxyribose (o=0) or ribose (o=1) with respective nucleobase	related to	C00215
related to	C00801	Same as	S00004, if p=0
same as	S00005[o,c,0]	Pathway	...
Pathway	...	Enzyme	...
Enzyme	...	Brite	...
Brite	...	URNs	...
URNs	...	KCF data	show
KCF data	show		

Figure 4.11: Substituents “(deoxy)Nucleoside” and “(deoxy)Nucleotide”. Note that for nucleosides (left) two different ways to formalize the parameters are shown.

Entry	S00006
Parameters	$o \in \{0, 1\}$ $S \in \begin{cases} [ACGT]^n & \text{if } o = 0 \\ [ACGU]^n & \text{if } o = 1 \end{cases}$
Name	Nucleic Acid
Composition	Sequence($S \Rightarrow c : C_5H_6O(2+o) HPO_3 S00001[c]$) OH
Reaction	...
Comment	sequence of phosphorylated deoxynucleosides (o=0) or phosphorylated nucleosides (o=1)
Same as	C00039[S], if o=0 C00046[S], if o=1
Pathway	...
Enzyme	...
Brite	...
URNs	...
KCF data	show

Figure 4.12: Nucleic Acid Entry. The notation $Sequence(S \Rightarrow c : C_5H_6O(2 + o) HPO_3 S00001[c]) OH$ reads: FOR EACH c in S : add $C_5H_6O(2+o) HPO_3$ and a nucleobase according to the value of c .

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

Entry	C00039
Parameters	$S \in [ACGT]^n$
Name	DNA; DNA _n ; DNA _{n+1} ; (Deoxyribonucleotide) _n ; (Deoxyribonucleotide) _m ; (Deoxyribonucleotide) _{n+m} ; Deoxyribonucleic acid
Composition	Sequence(S \Rightarrow c : C5H7O5 S00001[c] OH
Reaction	R00375 R00376 R00377 R00378 R00379 R00380 R00381 R00382
Comment	Sequence of deoxynucleotides (o=0) connected by phosphor groups Type: generic compound in reaction hierarchy
Same as	S00006[o=0,S]
Pathway	...
Enzyme	...
Brite	...
URNs	urn:miriam:cas:9007-49-2 urn:miriam:pubchem.substance:3341 urn:miriam:obo.chebi:16991 urn:miriam:jscd:J209.154B
KCF data	show

Entry	C00046
Parameters	$S \in [ACGU]^n$
Name	RNA; RNA _n ; RNA _{n+1} ; RNA(linear); (Ribonucleotide) _n ; (Ribonucleotide) _m ; (Ribonucleotide) _{n+m} ; Ribonucleic acid
Composition	Sequence(S \Rightarrow c : C5H7O6 S00001[c] OH
Reaction	R00435 R00436 R00437 R00438 R00439 R00440 R00441 R00442 R00443 R00444 R07282 R07640
Comment	Sequence of nucleotides (o=1) connected by phosphor groups Type: generic compound in reaction hierarchy
Same as	S00006[o=1,S]
Pathway	...
Enzyme	...
Brite	...
URNs	urn:miriam:pubchem.substance:3348
KCF data	show

Figure 4.13: Rule-based Compounds “DNA” and “RNA” Using our Suggested Fields for Composition and Crosslinking.

In the “Composition” field, we introduce the Sequence() tag, which allows for the calculation of an empirical formula if o and S are given. It reads as follows: For each character c in S the respective nucleotide with oxidation level o and one connecting phosphorus group is present, the last nucleic acid being terminated by an OH group. Depending on the choice of o , this molecule is either a DNA or an RNA, as stated in the “Same as” section. Following the structure used by KEGG we list the nucleic acids in a similar format (Figure 4.13). Note the altered “Composition”, “Same as”, and “URNs” rows, following the suggestions made before.

4.5.2.2 Reactions

In this section we show how reaction-class entries may be implemented in a rule-based metabolic database, allowing for both general annotations and the exact determination of formulas. We demonstrate how nucleic acid elongation may be described in a precise manner (Figures 4.14-4.15).

4.5 Towards Rule-Based Metabolic Databases

Entry	Rxxxx1
Parameters	$o \in \{0, 1\}$ $p \in \{0, 1, 2\}$ $c \in \begin{cases} \{A, C, G, T\} & \text{if } o = 0 \\ \{A, C, G, U\} & \text{if } o = 1 \end{cases}$
Name	Nucleotide Activation
Definition	Nucleotide + ATP \Leftrightarrow Nucleotide + ADP
Equation	S00005[o,p,c] + C00002 \Leftrightarrow S00005[o,p+1,c] + C00008
Same as	R00185, if o=1, p=0, c=A R02089, if o=0, p=0, c=A R01228, if o=1, p=0, c=G R01967, if o=0, p=0, c=G R00513, if o=1, p=0, c=C R01666, if o=0, p=0, c=C R00964, if o=1, p=0, c=U R01567, if o=0, p=0, c=T R00127, if o=1, p=1, c=A R01547, if o=0, p=1, c=A R00332, if o=1, p=1, c=G R02090, if o=0, p=1, c=G R00512, if o=1, p=1, c=C R01665, if o=0, p=1, c=C R00158, if o=1, p=1, c=U R02094, if o=0, p=1, c=T R00206, if o=0, p=2, c=A R00035, if o=1, p=2, c=G R00361, if o=0, p=2, c=G R00112, if o=1, p=2, c=C R00705, if o=0, p=2, c=C R00029, if o=1, p=2, c=U R00363, if o=0, p=2, c=T

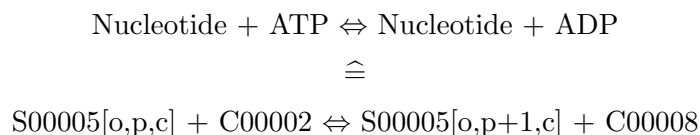
Figure 4.14: Rule-based Reaction “Nucleotide Activation”.

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

Entry	Rxxxx2
Parameters	$o \in \{0, 1\}$ $p \in \{1, 2\}$ $c \in \begin{cases} \{A, C, G, T\} & \text{if } o = 0 \\ \{A, C, G, U\} & \text{if } o = 1 \end{cases}$
Name	Nucleic Acid Elongation
Definition	Nucleic Acid + Nucleotide-Triphosphate \Leftrightarrow Diphosphate + Nucleic Acid
Equation	S00006[o,S] + S00005[o,c,3] \Leftrightarrow C00013 + S00006[o,S:c]
Same as	R00435, if o=1, c=A R00375, if o=0, c=A R00441, if o=1, c=G R00376, if o=0, c=G R00442, if o=1, c=C R00377, if o=0, c=C R00443, if o=1, c=U R00378, if o=0, c=T

Figure 4.15: Rule-based Reaction “Nucleic Acid Elongation”.

Nucleotide Activation Before nucleotides can be attached to the nucleic acid by a polymerase, they have to be activated by phosphorylation. This step is captured by the following reaction entry:



This entry uses the following parameters:

- o , the oxidation level of the nucleotide and distinguishes between ribonucleotides and desoxy-ribonucleotides, namely 0 for DNA and 1 for RNA;
- p , the number of phosphorus groups attached to the nucleotide before activation, and
- c , the specification of the base of the nucleotide.

The “Equation” field makes use of the substituent “Nucleotide” as described before and passes the parameters to it. By instantiating the parameters we get concrete reactions — like those already contained in KEGG. Note the changing phosphorylation parameter on the left hand side in contrast to the right hand side.

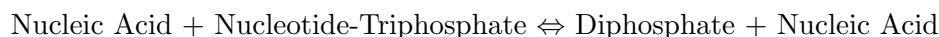
Figure 4.16 shows the example of the ATP:dTDP phosphotransferase. It basically reflects the KEGG entry, yet with a small addition: the “Same as” row reflects that we have an instantiation of the formerly described general reaction.

4.5 Towards Rule-Based Metabolic Databases

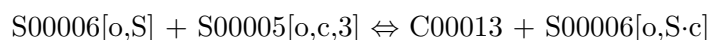
Entry	R02093
Name	ATP:dTDP phosphotransferase
Definition	ATP + dTDP \Leftrightarrow ADP + dTTP
Equation	C00002 + C00363 \Leftrightarrow C00008 + C00459
Same as	Rxxxx1[0,2,'T']
RPair	RP00003 C00002_C00008 main RP00142 C00363_C00459 main RP06834 C00002_C00459 trans
Enzyme	2.7.4.6
Pathway	rn00240 Pyrimidine metabolism rn01100 Metabolic pathways
Orthology	K00940 nucleoside-diphosphate kinase [EC:2.7.4.6]

Figure 4.16: Specific Reaction “ATP:dTDP Phosphotransferase” Instantiated from Rxxxx1 (4.14).

Elongation of Nucleic Acids Nucleic acids like RNA and DNA are polymers that can be elongated by chemical reactions. Usually, a triphosphate nucleotide is added to a nucleic acid and phosphate is released:



This process can now be formalized using parametrized species by the parametrized reaction Rxxxx2 (Figure 4.15).



where

- $c \in \{A, C, G, T, U\}$ is the abbreviation of the base used,
- S is the nucleic acid sequence before the elongation,
- $S \cdot c$ is the original sequence elongated by c ,
- o is the backbone oxidation level, and
- “3” in $S00005[o,c,3]$ means that the respective nucleotide is activated by 3 phosphate groups (yielding ATP, CTP etc. according to the value of c , cf. Figure 4.11)

This formalization improves the common one used in KEGG (and other databases) which only captures the polymer length.

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

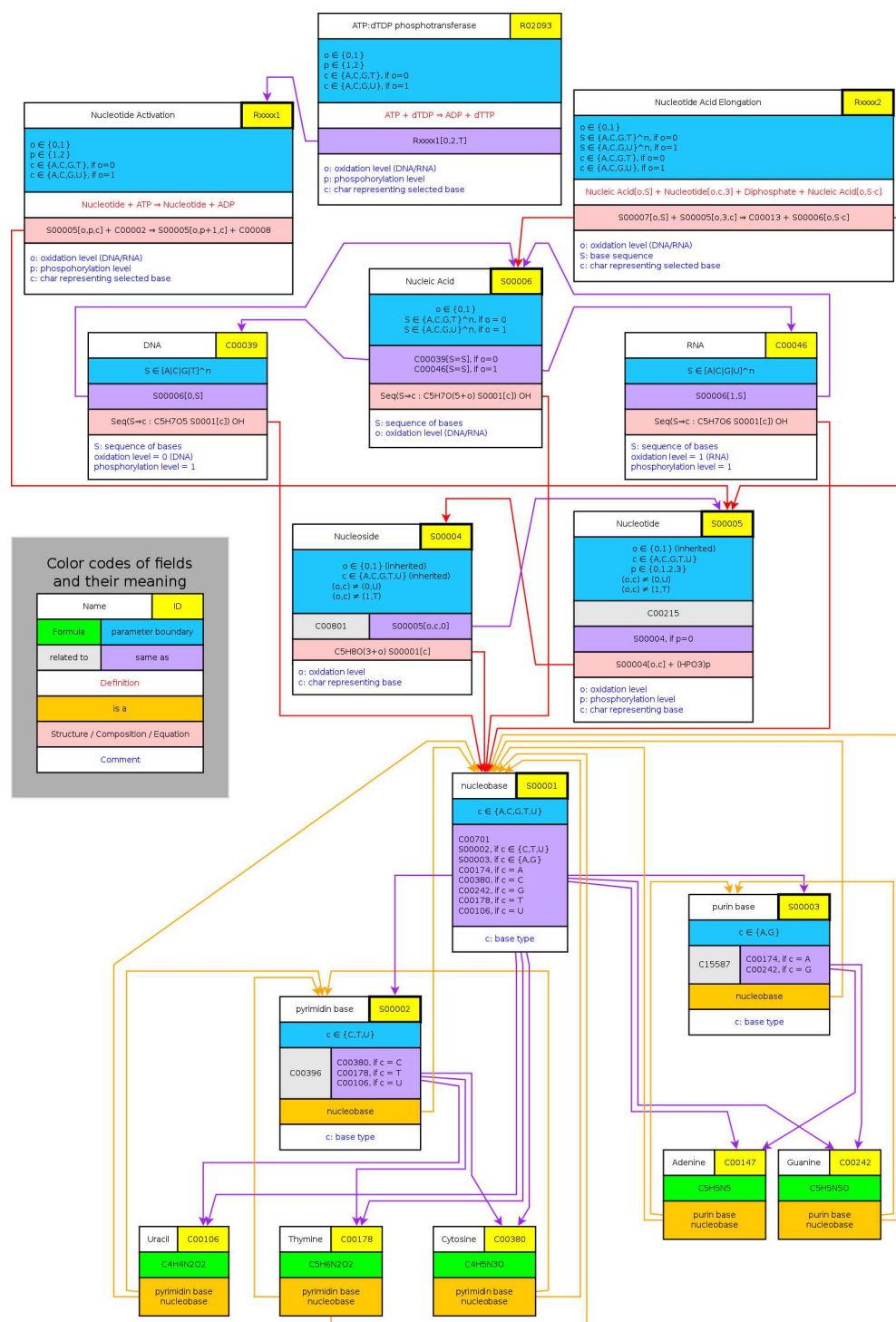


Figure 4.17: Relation of the Entries of the Example Presented to the Rule-based Data Base (Figures 4.8-4.15). Note that this graph is intended as an illustration of the relationships of the entries mentioned in this document. Thus, only a limited set of fields is shown, while the majority of informational fields in the KEGG database is omitted. The fields given in the chart should be easily applicable and can be used to complement existing fields and ontological data.

The interrelationship of all entries mentioned in the nucleic acid example is summarized in Figure 4.17. An overview on the reaction system is available at <http://tinyurl.com/rnaparts>.

4.6 Summary

We studied metabolic network database inconsistencies that are especially relevant for automatic network model reconstruction. To do so, we used a newly implemented tool (available from http://www.biosys.uni-jena.de/interaction_tools.html) which we applied to the KEGG database. For different classes of species and reaction inconsistencies we have presented short-term and long-term mitigation strategies. Only a small fraction of the inconsistencies are due to obvious input errors, which can, however, be easily corrected — reflecting the already high quality of the database. The largest fraction of inconsistencies is due to a lack of knowledge on part of the suppliers and to the combinatorial nature of certain species and reactions (cf. the example of nucleosides, nucleotides, and nucleic acids in Section Section 4.5).

To deal with these inconsistencies, we have suggested certain design principles for a future rule-based metabolic database. These principles can be summarized into three groups: *additional database fields*, *rule-based species and reactions*, and a *residue collection*.

4.6.1 Additional Database Fields

Currently, at least the “Comments” field in KEGG serves as a mere unstructured repository for all kinds of additional data such as publications, references to similar or “Same as” entries, notes regarding spontaneity or incompleteness of reactions, and other information. We suggest that dedicated rows are introduced for typical entry types including:

- boolean flags that capture spontaneity, incompleteness, and balance;
- references to literature and other data sources; and,
- internal references that distinguish between “Same as”, “Is a”, and “Related to” links.

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

This would add to a metabolic database features that are characteristic for ontologies. Apart from that, fields for additional structural representations, such as SMILES (c.f. Chapter 5.3.2)(66) and InCHI (c.f. Chapter 5.3.1)(29) strings, could be provided as they gradually become more common.

4.6.2 Rule-based Species And Reactions

In order to handle combinatorial species such as polymers and glycans together with the reactions they are involved in, we have suggested a rule-based approach employing parametrized species and parametrized reaction rules. For example, the elongation of a nucleic acid “Nucleic Acid + Nucleotide-Triphosphate \Leftrightarrow Diphosphate + Nucleic Acid” can now be formalized more elegantly using parametrized species such as “S00006[o,S] + S00005[o,c,3] \Leftrightarrow C00013 + S00006[o,S·c]”, which allows for the representation of a large (here: infinite) number of species and reactions in a condensed form. Rule-based species and reactions can then be used together with conventional species and reactions in a coherent way — which is, in turn, important to store special cases. A special case can instantiate a parametrized molecule (or reaction), while the parametrized molecule (or reaction) can be linked to the specific instance by the “Same as” link (see example entry C00226, above).

4.6.3 Substituent Collection

Many KEGG reactions contain generic compounds that encompass at least one **R** symbol in their formula. **R** symbols both on the substrate and the product side might refer either to the same or to diverging residues. Some of these residues are generic substance classes, whereas others constitute specific groups. Thus we suggest the introduction of a residue collection listing the classes of residues, their members, and for each of these members the respective instances of occurrence in substances and reactions. Each entry may make use of (parametrized) formulas and relations such as “Same as”, “Is a”, or “Related to”. A detailed example has been presented above (Section 4.5).

While we propose some strategies to deal with combinatorial complexity and formally inconsistent database entries, potentially reflecting incomplete biological knowledge, the same biological knowledge itself might at times be controversial. Still, this controversial data should be available in databases, and additional mechanisms have

to be introduced to cope with this. Moreover, some mitigation strategies eventually depend on the aim of the intended analysis.

4. INCONSISTENCY ANALYSIS ON THE KEGG DATABASE

5

RULE-BASED METABOLIC DATABASES

As described in Section 4, we need solutions for substance classes with non-unique formulas. A large part of those substances is made up by polymeric molecules which basically consist of a single or a few basic building blocks repeated multiple times, and some terminal groups. Most of them have a *regular* structure, which makes it possible to describe them based on simple rules using parameter variables.

This chapter examines the common classes of rule based molecules and describes a new formalism that allows rule-based annotation of the respective molecules. This is followed by existing frameworks that allow for a string representation of rulebased molecules. Finally, algorithms for basic calculations are examined in line with some examples and applications of the formalism.

5.1 Classes Of Rulebased Molecules

In the following, some possibilities to describe common cases of rule-based molecules are explained. These examples are grouped by classes of molecules:

1. Simplest case: **homopolymers** - polymers with variable length.

The most basic structural parameter of a polymer is its length. Simple polymers like alkanes, saturated monocarboxylic acids, or amylose can be described as the repetition of basic units (CH_2 in the case of alkanes and carboxylic acids, D-glucose in the case of amylose) attached to terminal groups. Therefore, the

5. RULE-BASED METABOLIC DATABASES

overall molecule formula can be determined if the number of repetitions n of the basic units is known. Examples for alkanes are methane, ethane, propane, butane, pentane... represented by $n = 1, 2, 3, 4, 5 \dots$ or acetic acid, propanoic acid, butanoic acid, pentanoic acid, hexanoic acid... for carboxylic acids, respectively.

2. **Heteropolymers** with several building blocks

Natural polymers are not limited to a single type of basic unit, but may have several kinds of building blocks. These can be as simple as hyaluronic acid, which consists of alternating units of D-glucuronic acid and D-N-acetylglucosamine, or as complex as nucleic acids consisting of at least 4 different units and proteins consisting of more than 20 building blocks with virtually infinite possible sequences.

3. **Attachments and charges**

There are some types of molecules for which it may be also adequate to model them as polymers, as they exhibit attachments with variable length. For example, phosphorylated nucleotides can be modeled in a polymer-like manner: the sugar-bound base can be seen as a terminal group attached to a phosphate-group polymer with lengths between zero (nucleoside) and 3 (nucleoside triphosphates). It should be noted that each attachment of such a phosphate group also introduces an additional negative charge at physiological pH values. Thus, both charge and number of phosphate groups may be represented by the length n .

4. **Branched and cross-linked structures**

All aforementioned polymer types possess a common property: they exhibit a linear structure. Nature also knows polymers with a branching structure. They range from those with simple branches as in the polysaccharide amylopectin to manifold crosslinked structures like vulcanized rubber.

5.2 Formalization Of Rule-based Reactions

In this section an appropriate formalism for rule-based reactions which has only been introduced informally before is derived. This formalism will be explained using three examples:

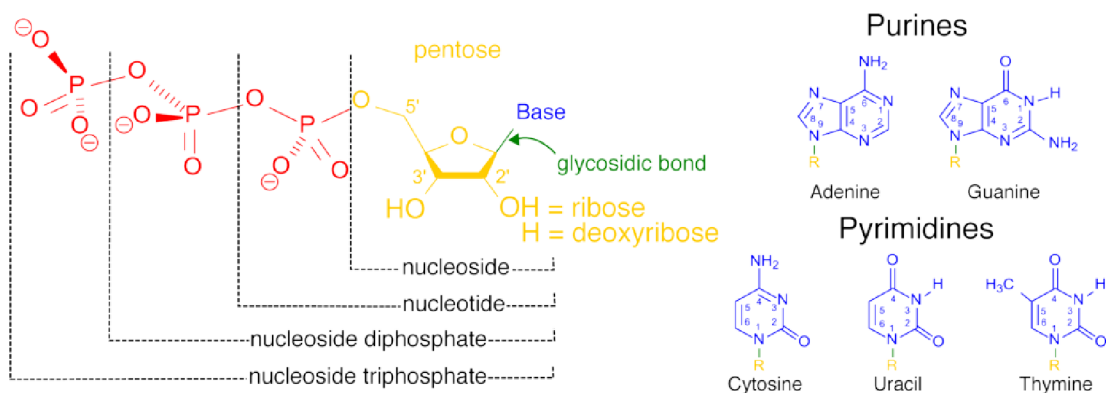


Figure 5.1: Generic scheme for nucleotides and common nucleotides

1. acyl-CoA thioesters, which will also be used for explanations in 5.4.3,
2. nucleotides, which have already been mentioned in 4.5.2.1 and
3. DNA, which also has been mentioned in 4.5.2.1.

5.2.1 Parametrized Molecules

We start by defining parametrized molecules. A fundamental property of parametrized molecules is that they are composed of invariant terminal atoms or groups on the one hand and variable substructures on the other hand.

Definition 1 A *substructure* is either a single atom or a molecular group that can be replaced by alternative atoms or molecular groups.

Considering our examples we start with the following situation:

1. For acyl-CoA thioesters, the two terminal groups are CH_3- and $-(CH_2)_2COS - CoA$. The repeated substructure is $-(CH_2)_2-$, c.f. Figure 5.6.
2. For the DNA molecule, the terminals can be an $-H$ atom and a phosphate group ($-PO_4^{3-}$). The intercalated substructures are adenine, cytosine, guanine and thymine (Figure 5.1) in varying amounts.
3. A nucleotide can be seen as a terminal sugar with two variable substructures: a nucleic base, either adenine, cytosine, guanine, thymine or uracil, and one to three phosphate groups (Figure 5.1).

5. RULE-BASED METABOLIC DATABASES

Depending on the molecule type, either the type or the amount of substructures of a molecule can be described as variables. We call these variables the **parameters** of parametrized molecules. Note that a certain molecule (chemical formula) may belong to several types of parametrized molecules. For a parametrized molecule, a parameter can either denote the number of repetitions of a certain substructure, or characterize which of the possible substructures appears at a given variable position. In the latter case, a single substructure is represented by a symbol, daisy-chained substructures are represented as a sequence of symbols, called **strings**. Note that a single symbol can be interpreted as a string with the length one.

Definition 2 *We define Σ as the set of all symbols that can be used to represent specific substructures. Furthermore, Σ^* is meant to denote the Kleene closure of Σ , i.e. the set of all strings built upon Σ .*

1. Acyl-CoA thioesters form a class of molecules that differ in their length or, more precisely, in the number of repetitions of the $-(CH_2)_2-$ substructure. Accordingly, the different members of this class are discriminated by a single numerical parameter.
2. The DNA molecules with the sequences 'GATTACA' and 'ATTAC' are exemplary instances of an anticipated DNA molecule class. Each instance is described by a single string parameter.
3. Nucleotides form a class of molecules whose members can differ in both their phosphorylation level and the nucleic base bound. Hence, this class has one numerical parameter for the phosphorylation level and one string-type parameter to represent the base.

Let n and s denote the count of numerical and string-type parameters of a parametrized molecule, respectively. Accordingly, the domains \mathcal{P} of the parameters of a parametrized molecule are a subset of the Cartesian product of \mathbb{N}^n and $(\Sigma^*)^s$:

Definition 3 *Let \mathcal{F} be the set of all possible chemical sum formulas. Let \mathcal{P} be the domain of a set of parameters. We define the set \mathcal{M} of **parametrized molecule classes** as follows:*

$$\mathcal{M} = \{\mathcal{P} \rightarrow \mathcal{F} : \mathcal{P} \subseteq \mathbb{N}^n \times (\Sigma^*)^s; n, s \in \mathbb{N}\}$$

5.2 Formalization Of Rule-based Reactions

Let $n + s = m$. Each element $M \in \mathcal{M}$, $M : \mathcal{P}_1 \times \mathcal{P}_2 \times \dots \times \mathcal{P}_m \rightarrow \mathcal{F}$ together with the definitions of all parameter domains \mathcal{P}_i is called **signature** of a parametrized molecule class.

Note that, in principle, numerical and string type parameters could be in a mixed order. The separation into *numerical*-first and *string-type*-last has been chosen w.l.o.g. to make explanation easier and could be altered arbitrarily.

1. Acetyl-CoA, Butanoyl-CoA, Hexanoyl-CoA, and Octanoyl-CoA are examples of molecules belonging to the class of parametrized acyl-CoA thioester molecules. The signature of this class reads

$ACT : allowed-lengths \rightarrow \mathcal{F}; allowed-lengths \subseteq \mathbb{N}$.

The aforementioned members can be represented by the parametrized expressions $ACT(1)$, $ACT(2)$, $ACT(3)$, and $ACT(4)$.

2. The signature of the class of DNA molecules is

$DNA : seq \rightarrow \mathcal{F}; seq \subseteq \{'A', 'C', 'G', 'T'\}^*$.

The DNA molecules with the sequences 'GATTACA' and 'ATTAC' are represented as $DNA('GATTACA')$ and $DNA('ATTAC')$.

3. $Nt : phos \times base \rightarrow \mathcal{F}; phos = \{1, 2, 3\}$, $base = \{'A', 'C', 'G', 'T', 'U'\}$ is the signature of parametrized nucleotide molecules. Guanosine diphosphate, Uridine monophosphate and Adenosine triphosphate are three instances that can be written as $Nt(2, 'G')$, $Nt(1, 'U')$ and $Nt(3, 'A')$. Each of them has one numeric parameter representing the phosphorylation level and one string type parameter that represents the nucleic base of the nucleotide.

Parameterized molecules with a signature that only contains numeric variables (i.e. $M : \mathbb{N}^n \rightarrow \mathcal{F}$) can directly be translated to CurlySMILES strings: The CurlySMILES annotation of acyl-CoA thioesters is, for instance, given in Figure 5.7, on the top.

5.2.2 Formalization Of Reactions

In this section, we derive a formalism for the application of the previously defined parametrized molecules within rule-based reactions. Again, to demonstrate this approach, we will make use of three (simplified) examples:

5. RULE-BASED METABOLIC DATABASES

1. $ACT(2) + 2CH_2 \rightleftharpoons ACT(3)$, namely Butanoyl-CoA + $2CH_2 \rightleftharpoons$ Hexanoyl-CoA
2. $Nt(2,'C') + Nt(3,'A') \rightleftharpoons Nt(3,'C') + Nt(2,'A') \hat{=} CDP + ATP \rightleftharpoons CTP + ADP$
3. $DNA('ATTA') + Nt(3,'C') \rightleftharpoons DNA('ATTAC') + PP_i$ (c.f. Figure 4.1b)

These three examples are instances of three very different classes of reactions which will be explained subsequently.

The first reaction is the elongation of an acyl-CoA thioester. We can get very similar but nonetheless different reactions by changing the values of the length parameters of the educt. This change would also be reflected in a change of the length parameter of the product. It stands to reason that these reactions, having different educt and product instances but sharing the same intrinsic structure, are to be grouped into one reaction class.

The second reaction describes the transfer of a phosphate group from ATP to CDP to form ADP and CTP. In this case, both the phosphate group donating base and the accepting base could be replaced by another base and we would get another reaction with the same structure. Again, it is plausible to put all reactions that transfer a phosphate group from a tri-phospho-nucleotide to a di-phospho-nucleotide into one bin, forming another specific reaction class.

The last reaction displays the attachment of a single nucleotide to a given DNA strand, thereby elongating it. Once more, the DNA sequence as well as the added base could be altered, producing another instance of an anticipated *DNA elongation* reaction class.

Altogether, there are two main differences between these classes:

1. the number of educts and products and
2. the type of each educt and each product.

Definition 4 Let \mathcal{R} be the set of all reactions. Let $R_{e,p} = \mathcal{M}^e \times \mathcal{M}^p \subset \mathcal{R}$; $e, p \in \mathbb{N}$ be the set of all reactions with e educts and p products. Then each $R = ((M_1, M_2, \dots, M_e), (M_{e+1}, M_{e+2}, \dots, M_{e+p})) \in R_{e,p}$ is a set of reactions with fixed molecule classes at each position called a **prototype class**.

A prototype class fixes the molecule class occurring at each position of the educt and product tuples, but does not constrain any educt or product parameter value. A specific

5.2 Formalization Of Rule-based Reactions

rule-based reaction class is defined by adding such constraints to a certain prototype class. These constraints are then defined by a mapping function belonging to the defined reaction class. In other words: from the prototypic set of reactions R only a fraction is realized. Those reactions which are actually realized are defined by a corresponding function F^R that maps allowed educt instances to allowed product instances.

Definition 5 *The signature of a rule-based reaction class is defined as (R, F^R) with*

$$R \in R_{e,p}; \quad e, p \in \mathbb{N}$$

$$F^R : (M_{1|P_1} \times M_{2|P_2} \times \dots \times M_{e|P_e}) \rightarrow (M_{e+1|P_{e+1}} \times M_{e+2|P_{e+2}} \times \dots \times M_{e+p|P_{e+p}}),$$

where $M_{i|P_i}$ is a molecule class with its parameter ranges restricted to certain subsets defined by P_i .

1. The signature of the acetyl-CoA elongation reaction class is

$$R_{act-elong} = ((ACT_1, CH_2), (ACT_2)) \quad \text{with}$$

$$ACT_1 \quad : \quad length_1 \rightarrow \mathcal{F} \quad (5.1)$$

$$CH_2 \quad : \quad \{\} \rightarrow \mathcal{F} \quad (5.2)$$

$$ACT_2 \quad : \quad length_2 \rightarrow \mathcal{F} \quad (5.3)$$

$$length_1 = length_2 = \mathbb{N} \quad (5.4)$$

$$F_{R_{act-elong}} \quad : \quad length_1 \times \{\} \rightarrow length_2 \quad (5.5)$$

$$F_{R_{act-elong}}(l) = l + 1, \quad l \in length_1 \quad (5.6)$$

2. The signature of the nucleotide activation is

$$R_{activation} = ((Nt_1, Nt_2), (Nt_3, Nt_4)) \quad \text{with}$$

5. RULE-BASED METABOLIC DATABASES

$$Nt_1 : phos_1 \times base_1 \rightarrow \mathcal{F} \quad (5.7)$$

$$Nt_2 : phos_2 \times base_2 \rightarrow \mathcal{F} \quad (5.8)$$

$$Nt_3 : phos_3 \times base_3 \rightarrow \mathcal{F} \quad (5.9)$$

$$Nt_4 : phos_4 \times base_4 \rightarrow \mathcal{F} \quad (5.10)$$

$$phos_1 = \{1, 2\} \quad (5.11)$$

$$phos_2 = \{3\} \quad (5.12)$$

$$phos_3 = \{2, 3\} \quad (5.13)$$

$$phos_4 = \{2\} \quad (5.14)$$

$$base_1 = base_3 = \{'A', 'C', 'G', 'T', 'U'\} \quad (5.15)$$

$$base_2 = base_4 = \{'A'\} \quad (5.16)$$

$$F_{R_{activation}} : (phos_1 \times base_1) \quad (5.17)$$

$$\rightarrow (phos_3 \times base_3) \times (phos_4 \times base_4)$$

$$F_{R_{activation}}(p, b) = ((p + 1, b), (2, 'A')), p \in phos_1, b \in base_1 \quad (5.18)$$

3. The signature of the DNA elongation is

$$R_{dna-elong} = ((DNA_1, Nt), (DNA_2)) \text{ with}$$

$$DNA_1 : seq_1 \rightarrow \mathcal{F} \quad (5.19)$$

$$Nt : phos \times base \rightarrow \mathcal{F} \quad (5.20)$$

$$DNA_2 : seq_2 \rightarrow \mathcal{F} \quad (5.21)$$

$$phos = \{3\} \quad (5.22)$$

$$base = \{'A', 'C', 'G', 'T'\} \quad (5.23)$$

$$seq_1 = seq_2 = base^* \quad (5.24)$$

$$F_{R_{dna-elong}} : seq_1 \times base \rightarrow seq_2 \quad (5.25)$$

$$F_{R_{dna-elong}}(s, b) = (s \circ b), s \in seq_1, b \in base \quad (5.26)$$

The symbol “ \circ ” denotes the concatenation of two string type parameters.

5.2.3 Mapping Between Educts And Products

Considering the signatures of the examples shown in the previous section we see that, in addition to the domains of all reactant parameters, the signature contains a mapping function F_R . This mapping function determines the interdependency between educts and products and has some restrictions described in the following:

Reactions without educts or without products are widely used to model inflows or outflows in biochemical models. Keeping aside nuclear chemistry, in real-world systems there are no reactions that create or destroy matter due to the law of conservation of mass. This implies that

- in a classical chemical context neither the educt nor the product set may be empty and that
- relations between products and educts are always linear, as is the corresponding mapping function.

In other words: the parameters of the product site have to be constrained in linear dependence on the educt side parameters.

From a chemical point of view, a reaction takes a non-empty collection of molecules and transforms it into another non-empty collection of molecules. In a rule-based reaction class at least two of these molecules are parametrized. Considering the example $Nt(p_1, b_1) + Nt(3, A') \rightleftharpoons Nt(p_1 + 1, b_1) + Nt(2, A')$ we see that each group of molecules can contain several instances of one and the same class of molecules.

Considering the possible representations of molecules, we can have the following six basic transformation types, where S represents sequences of symbols and n represents integers.



Of course, these variables can be combined to form more complex reaction patterns, different strings can be build upon another alphabet and different integer variables can have their own ranges. Remember that the integer n represents the quantity of a

5. RULE-BASED METABOLIC DATABASES

certain repetitive substructure and the string S represents a specific concatenation of certain substructures. If a variable appears several times within one reaction equation, it denotes the same quantity of the same substructure or the same sequence of the same substructures for each occurrence in the equation. By the law of conservation of mass, the following operations are not permitted:

$$\dots + M_1(\dots, n, \dots) \leftrightarrow M_2(\dots, c \cdot n, \dots) + \dots \quad (5.30)$$

$$\dots + M_1(\dots, S, \dots) \leftrightarrow M_2(\dots, S \circ S \circ \dots, \dots) + \dots \quad (5.31)$$

If they were, the corresponding reactions would create or destroy copies of their respective substructures.

Thus, we can derive the following rules to describe the interdependency of educts and products: Each numerical product parameter may potentially depend on each numerical educt parameter and each string type product parameter may depend on each string type educt parameter. Additionally, numerical parameters can be transformed into symbols/strings and vice versa.

Let e and p be natural numbers ($e, p \in \mathbb{N}$).

Let $\forall_{i \in \{1, 2, \dots, e+p\}} M_i : \mathcal{P}_{i,1} \times \text{Par}_{i,2} \times \dots \times \mathcal{P}_{i,m_i} \rightarrow \mathcal{F}$ be the signatures of different parametrized molecules. Let R be a reaction with the signature (R, F^R) ,

$$R = ((M_1, M_2, \dots, M_e), (M_{e+1}, M_{e+2}, \dots, M_{e+p})),$$

$$F^R : \begin{pmatrix} \mathcal{P}_{1,1} & \times & \dots & \times & \mathcal{P}_{1,m_1} \\ \times & \mathcal{P}_{2,1} & \times & \dots & \times & \mathcal{P}_{2,m_2} \\ \times & \dots & \times & \dots & \times & \dots \\ \times & \mathcal{P}_{e,1} & \times & \dots & \times & \mathcal{P}_{e,m_e} \end{pmatrix} \rightarrow \begin{pmatrix} \mathcal{P}_{e+1,1} & \times & \dots & \times & \mathcal{P}_{e+1,m_{e+1}} \\ \times & \mathcal{P}_{e+2,1} & \times & \dots & \times & \mathcal{P}_{e+2,m_{e+2}} \\ \times & \dots & \times & \dots & \times & \dots \\ \times & \mathcal{P}_{e+p,1} & \times & \dots & \times & \mathcal{P}_{e+p,m_{e+p}} \end{pmatrix}$$

Let $p_{k,l} \in \mathcal{P}_{k,l}$ be the l^{th} parameter of the k^{th} educt molecule. Let $F_{i,j}^R$ be the function that describes the j^{th} parameter of the i^{th} product molecule. Then, for each numerical product parameter

$$F_{i,j}^R(p_{1,1}, \dots, p_{1,m_1}, \dots, p_{e,1}, \dots, p_{e,m_e}) = c_{i,j} + \sum_{k=1}^e \sum_{l=1}^{m_k} c_{i,j,k,l} \cdot p_{k,l} \text{ holds.}$$

Altogether, this formula describes that all numeric educt parameters are combined for each educt, weighted with $c_{i,j,k,l} \in \mathbb{N}$. For the character and string type parameters, all typical string operations like concatenation ($S_1 \circ S_2$) and slicing are allowed.

5.3 Annotation Frameworks For Rulebased Reactions

Furthermore, conversions of strings from one realm (underlying alphabet) to another realm are possible:

$RNA('AUG') + \dots \rightarrow Triplet('M') + \dots$ is a fictional example that represents an RNA-snippet becoming relevant as methionine-coding t-RNA. The only boundary for string operations is the inviolability of the law of mass conservation, expressed in equations 5.30 and 5.31.

5.2.4 Reactivity Check

For algorithms implementing an artificial chemistry or predicting properties of a chemical system, a crucial step is to check whether a certain reaction $R = (E, P)$ can perform on a given set \hat{E} of educt molecules. For non-rule-based chemical systems, this step is accomplished simply by checking whether the given molecules fit the reaction's educts. For rule-based systems, not only the educt set has to contain the right classes of molecules, but also the validity of all parameters of educts and products has to be checked.

Thus, checking the feasibility of a reaction on a given molecule multiset \hat{E} is a two step process:

1. matching educt class types

The first condition which a set of educts has to meet is the following:

$\exists \pi : \hat{E} \rightarrow \hat{E} : \hat{E}' = \pi(\hat{E}) \in E$ - there is a permutation \hat{E}' of \hat{E} , so that each position of the educt part E of the signature is matched by tuple elements in \hat{E} .

2. checking product parameter validity

$F^R(\hat{E}') \in P$ - the parameters calculated for the products by applying the mapping function have to be within the product parameter range constraints.

5.3 Annotation Frameworks For Rulebased Reactions

This section deals with approaches to give molecular formulas a textual representation. Such string representations of formulas are needed to store molecular information in databases and search for molecules in online repositories. They are recognized by various programs. There are a few formats for such formula encoding texts, of which InChI and SMILES are the most important.

5. RULE-BASED METABOLIC DATABASES

5.3.1 InChI

The International Chemical Identifier is an open string representation standard for chemical structures developed by the International Union of Pure and Applied Chemistry (IUPAC, (29)). It is a system and method to convert chemical structures and formulas to linear text strings using a subset of characters of ASCII. This representation can be utilized to store such strings in databases and use them for computational purposes. Here, each molecule is mapped to a specific, unique InChI string, which proves a huge advantage in view of database usage. A major drawback of InChI is, that there currently is no approach allowing for the short-hand generic annotation of substances with repetitive substructures.

5.3.2 SMILES

SMILES is the abbreviation for *Simplified Molecular Input Line Entry Specification* and was developed in the late 1980s by Arthur and David Weininger (66). Similar to InChI it maps molecules to their respective linear textual representations which can be used in computer-aided research. Following the proprietary original specification, an open standard has been developed (33). Although both, i.e. the original paper and openSMILES, proposed a possibility for the annotation of residue groups and polymers, this task can be realized even better with the extension CurlySMILES (18): This paper defines rules for the annotation of non-covalent bonds, the attachment of biomolecules and the representation of repetitive units, i.e. polymers. The CurlySMILES representation will be used in some of the following explanations.

5.3.3 BNGL

The BioNetGen Language (BNGL)(19) is a structured formal language allowing for a detailed description of biochemical models by a rule-based formalism. This formalism captures parameters, molecule types, seed species, reaction rules, observables, and actions (27). It makes possible to define binding sites, states, interactions, and components of molecules as well as the respective reaction rules acting on these. BNGL allows for the modeling of biochemical networks on a detailed, yet abstract level and can be used with the BioNetGen framework to perform model simulations. Unfortunately,

BNGL tends to be barely human-readable and might be to much of a good thing for the purpose of annotating molecules and reactions in biochemical databases.

5.4 Homopolymers - Polymers Of Variable Length

Polymers which merely differ in the number of their basic units are rather widespread in nature. A quick search in our local excerpt from the KEGG database (cf. Section 4.3.2) reveals, amongst others, the following types:

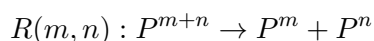
- Sugars and their derivates. There are plenty of types of sugar molecules, which may be concatenated to form linear chains of equal blocks.
- Nucleic acids. Although they may only form a small fraction of the existing nucleotides with a less important role, there are poly-X-Nucleotides, with X being one out of A, C, G, T. These Polymers are captured by the KEGG entries C00549, C00895, C01445, C02072.
- Peptides. Similar to nucleotides, aminoacids may be concatenated to form homopolymers like polyarginine or polyproline (KEGG C01952 and C01843, respectively).
- Derivates of hydrocarbons like isoprene, vinyl, prenyl and others. Example: dolichol phosphate, C00110
- Phosphate polymers (C00404, C02174)

These classes are common and comprise repetitive elements on the sub-molecular level. Of course, there may be even more examples at higher levels, where molecules of the same type and structure aggregate to form polymers such as f-actin. Yet, in the spotlight of reaction systems, these supramolekular polymers are out of scope. In the following, we demonstrate two abstract approaches dealing with the question which calculations can be easily performed on such polymers, followed by a real-world example.

5. RULE-BASED METABOLIC DATABASES

5.4.1 Example 1: Interval Chemistry

This section presents an example of a degradation network model and how the CLOSURE of the network can be calculated without expanding the network to all of its participating substances. The molecules of the networks are polymers P^k , where k denotes the length of the polymer. The only reaction (besides inflows and outflows) has the following form:



This reaction is constrained by the allowed sizes of the fragments $m \in \{m_{min}, \dots, m_{max}\}$ and $n \in \{n_{min}, \dots, n_{max}\}$. For an inflow set given by the respective intervals, the possible outflows and intermediates can be calculated without resolving the interval to molecule instances. For the subsequent algorithm descriptions, the following interval values were chosen:

- $m \in \{10, \dots, 30\}$ for the first reaction product
- $n \in \{15, \dots, 40\}$ for the second reaction product
- $k \in \{65, \dots, 80\}$ for the inflow

The following pseudo code snippets have been implemented and tested in python using the IP[y]: Notebook¹. The source codes can be found in Supplement 9.

5.4.1.1 Approximative Algorithm

A straightforward method to find the intervals of length values of possible products (\tilde{m} and \tilde{n}) is given by the following approximative steps:

1. Initialize:

$$\begin{aligned}\tilde{m}_{min} &:= m_{min}, & \tilde{n}_{min} &:= m_{min}, \\ \tilde{m}_{max} &:= m_{max}, & \tilde{n}_{max} &:= n_{max}, \\ \delta_m &:= m_{max} - m_{min}, & \delta_n &:= n_{max} - n_{min}\end{aligned}$$

2. abort, if $k_{max} < m_{min} + n_{min}$ or $m_{max} + n_{max} < k_{min}$: inflow out of range

3. determine minimum value of \tilde{m}_{min} :

$$\text{while } \begin{cases} \tilde{m}_{min} + n_{max} < x_{min} & : \text{ halve } \delta_m, \text{ increase } \tilde{m}_{min} \text{ by } \delta_m \\ \tilde{m}_{min} + n_{max} > x_{min} & : \text{ halve } \delta_m, \text{ decrease } \tilde{m}_{min} \text{ by } \delta_m \end{cases}$$

¹<http://ipython.org/notebook.html>

5.4 Homopolymers - Polymers Of Variable Length

4. determine minimum value of \tilde{n}_{min} :
 while $\begin{cases} \tilde{n}_{min} + m_{max} < x_{min} & : \text{ halve } \delta_n, \text{ increase } \tilde{n}_{min} \text{ by } \delta_n \\ \tilde{n}_{min} + m_{max} > x_{min} & : \text{ halve } \delta_n, \text{ decrease } \tilde{n}_{min} \text{ by } \delta_n \end{cases}$
5. reset δ_m and δ_n to the values given in 1st step
6. approximate maximum value of \tilde{m}_{max} :
 while $\begin{cases} \tilde{m}_{max} + n_{min} < x_{max} & : \text{ halve } \delta_m, \text{ increase } \tilde{m}_{max} \text{ by } \delta_m \\ \tilde{m}_{max} + n_{min} > x_{max} & : \text{ halve } \delta_m, \text{ decrease } \tilde{m}_{max} \text{ by } \delta_m \end{cases}$
7. approximate maximum value of \tilde{n}_{max} :
 while $\begin{cases} \tilde{n}_{max} + m_{min} < x_{max} & : \text{ halve } \delta_n, \text{ increase } \tilde{n}_{max} \text{ by } \delta_n \\ \tilde{n}_{max} + m_{min} > x_{max} & : \text{ halve } \delta_n, \text{ decrease } \tilde{n}_{max} \text{ by } \delta_n \end{cases}$
8. return $\{m_{min}, \dots, m_{max}\}$ and $\{n_{min}, \dots, n_{max}\}$ as sets of actually generated products.

Steps 1 to 8 calculate the *direct* products of a given inflow set for a reaction system. Their worst case execution time is $O(\log[\max(|n|, |m|)])$, where $|m| = m_{max} - m_{min}$ denotes the length of the interval. To calculate the scope or closure of a given input set, these steps have to be iterated.

5.4.1.2 Graphical Analysis

Here, we show how this problem can be solved geometrically. Figure 5.2 shows the initial situation as follows: The inflow of molecules P^k is given as the area between the purple, dashed diagonal lines. The range of possible products P^m is indicated on the ordinate as the area between the blue, dashed horizontal lines. Possible products P^n are given on the abscissa, their range is drawn with red, dashed vertical lines.

The rectangular area between the horizontal and the vertical dashed lines is the range of possible reactions of this system, shaped by the possible values of m and n . The diagonal (pink) shaded bar in the right corresponds to substrates that cannot be processed. The overlap (green) of the reaction space with the range of inflows is the set of actually performing reactions. From there, the actually produced sets of products are projected onto the abscissa and ordinate (horizontal and vertical blue shaded bars). The resulting substances are given by $m \in \{25, \dots, 30\}$ and $n \in \{35, \dots, 40\}$, and are projected (yellow diagonal bars) onto the reaction space. A part of these products can be processed further, indicated by those parts of the yellow bars that overlap with

5. RULE-BASED METABOLIC DATABASES

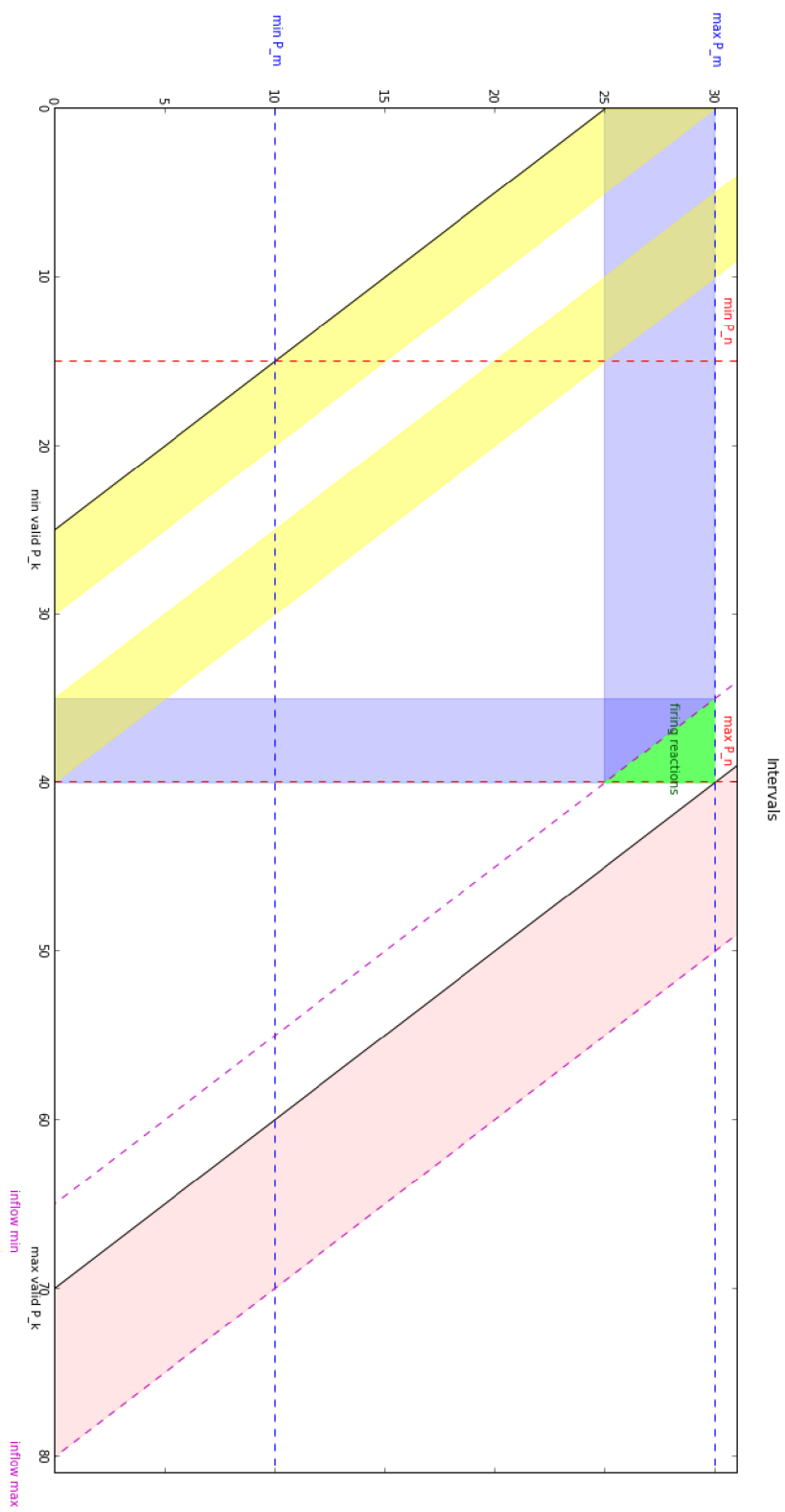


Figure 5.2: Visualization of the relation of inflows and outflows: first iteration

The green triangle represents the overlap between the actual inflows (dashed diagonal) and the processable components. Products are projected to ordinate and abscissa (horizontal and vertical blue bars). The graphical algorithm is described in Section 5.4.1.2.

5.4 Homopolymers - Polymers Of Variable Length

the reaction space rectangle. To find the final products of the reaction system, the outcome has to be traced further. Figure 5.3 shows the projection of A_m and A_n onto the reactive area (yellow diagonal areas). The new green areas (cp. Figure 5.2) are formed by the overlap of products of the previous step with the range of possible reactions. Again, the respective products are projected onto the ordinate (P_m) and abscissa (P_n). Finally the values of all produced substances are projected diagonally onto the reaction space, until we get the leftmost area of inert products (pink shaded diagonal bars). The diagonal bars altogether represent the CLOSURE of this reaction system:

$$C(\{65, \dots, 80\}) = \{10, \dots, 30\} \cup \{35, \dots, 40\} \cup \{65, \dots, 80\}$$

5.4.1.3 Analytic Constant Time Algorithm

While the sequential approximative algorithm (cp. 5.4.1.1) shows an intuitive way to calculate the direct products of a given inflow in logarithmic time, the graphical analysis indicates that the calculation of each iteration should be possible within a constant time. To do so, one just needs to find the correct intersection points of the intervals and do a proper projection.

This is accomplished by the following structured algorithm, where \square_{min} and \square_{max} represent the lower and upper boundaries of the interval $\square \equiv \{\square_{min}, \dots, \square_{max}\}$:

1. module - calculation of processable compounds: $O(1)$
the interval for processable compounds is just $[m_{min} + n_{min}, m_{max} + n_{max}]$.
2. module - calculate the processed subset of k : $O(1)$
 - calculate the interval of processable compounds as $p \equiv \{p_{min}, \dots, p_{max}\}$ using the 1st module
 - if $k_{min} > p_{min}$ then $p_{min} := k_{min}$
 - if $k_{max} < p_{max}$ then $p_{max} := k_{max}$
 - return p
3. module - calculate products corresponding to interval k : $O(1)$

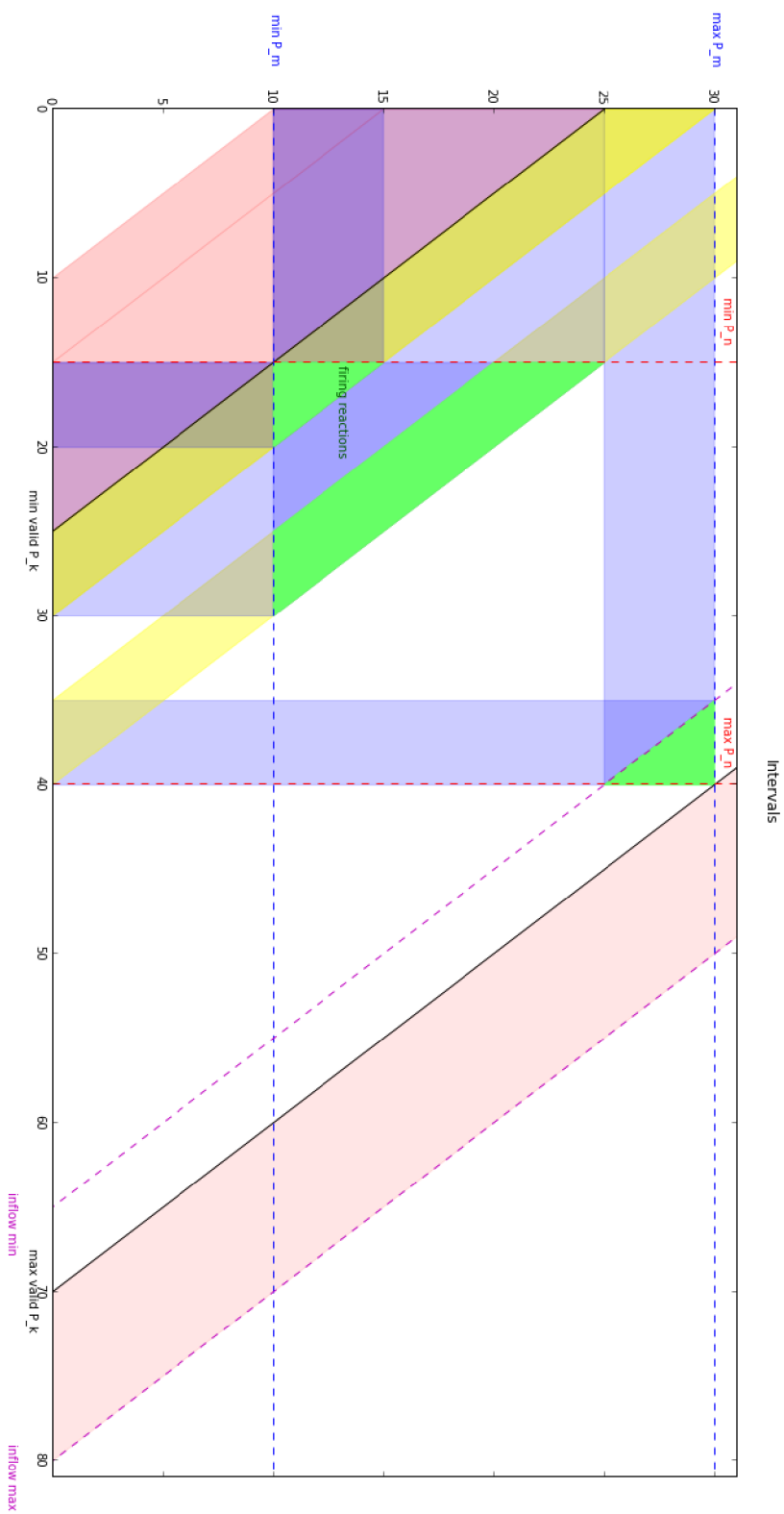


Figure 5.3: Visualization of the relation of inflows and outflows: final. The projected products have been mapped back (yellow diagonal bars) to the area of processable compounds and new products are formed (blue horizontal and vertical bars). The graphical algorithm is described in Section 5.4.1.2. The figure has been created with the code given in Supplement 9

- Initialize:

$$\begin{aligned} \tilde{m}_{min} &:= m_{min}, & \tilde{n}_{min} &:= m_{min}, \\ \tilde{m}_{max} &:= m_{max}, & \tilde{n}_{max} &:= n_{max}, \\ \delta_m &:= m_{max} - m_{min}, & \delta_n &:= n_{max} - n_{min} \end{aligned}$$

- calculate the processed subset of k as p using 2nd module
- abort, if p is empty: unfeasible inflow
- if $m_{min} + n_{max} \leq p_{min}$ then $\tilde{m}_{min} := p_{min} - n_{max}$
- if $m_{min} + n_{max} > p_{max}$ then $\tilde{n}_{max} := p_{max} - n_{min}$
- if $m_{max} + n_{min} \leq p_{min}$ then $\tilde{n}_{min} := p_{min} - m_{max}$
- if $m_{max} + n_{min} > p_{max}$ then $\tilde{m}_{max} := p_{max} - n_{min}$
- return \tilde{m}, \tilde{n} - merged if overlapping

5.4.2 Example 2: Coupling Of Equal Chains

For networks of the form $n \cdot A^m \Leftrightarrow A^{n \cdot m}$ (where the raised variable determines the polymer length) we can do the following generic calculations. Some exemplary structures of such networks including several iterations (depth, “d-level”) of linkage reactions are sketched in Figure 5.4. Two representations of generic building blocks for such networks are shown in Figure 5.5. To convert from explicit structures as given in Figure 5.4 to a canonic form (Figure 5.5, bottom), the stoichiometric parameters c (cycling) and e (efflux) have to be determined. In the drawing, it can be seen that every d-level consists of n times the number of reactions than the previous d-level, increased by one. So the number z of reactions in the k th d-level is given by

$$z_k = n \cdot z_{k-1} + 1 = \frac{n^k - 1}{n - 1}$$

For every d-level, there is one efflux-reaction, meaning that the fraction of component efflux in Figure 5.5, bottom, is

$$e = \frac{1}{z_k} = \frac{n - 1}{n^k - 1}$$

As the total number of products for each of these reactions is one, with $c + e = 1$ we get

$$c = 1 - e = \frac{n^k - n}{n^k - 1}$$

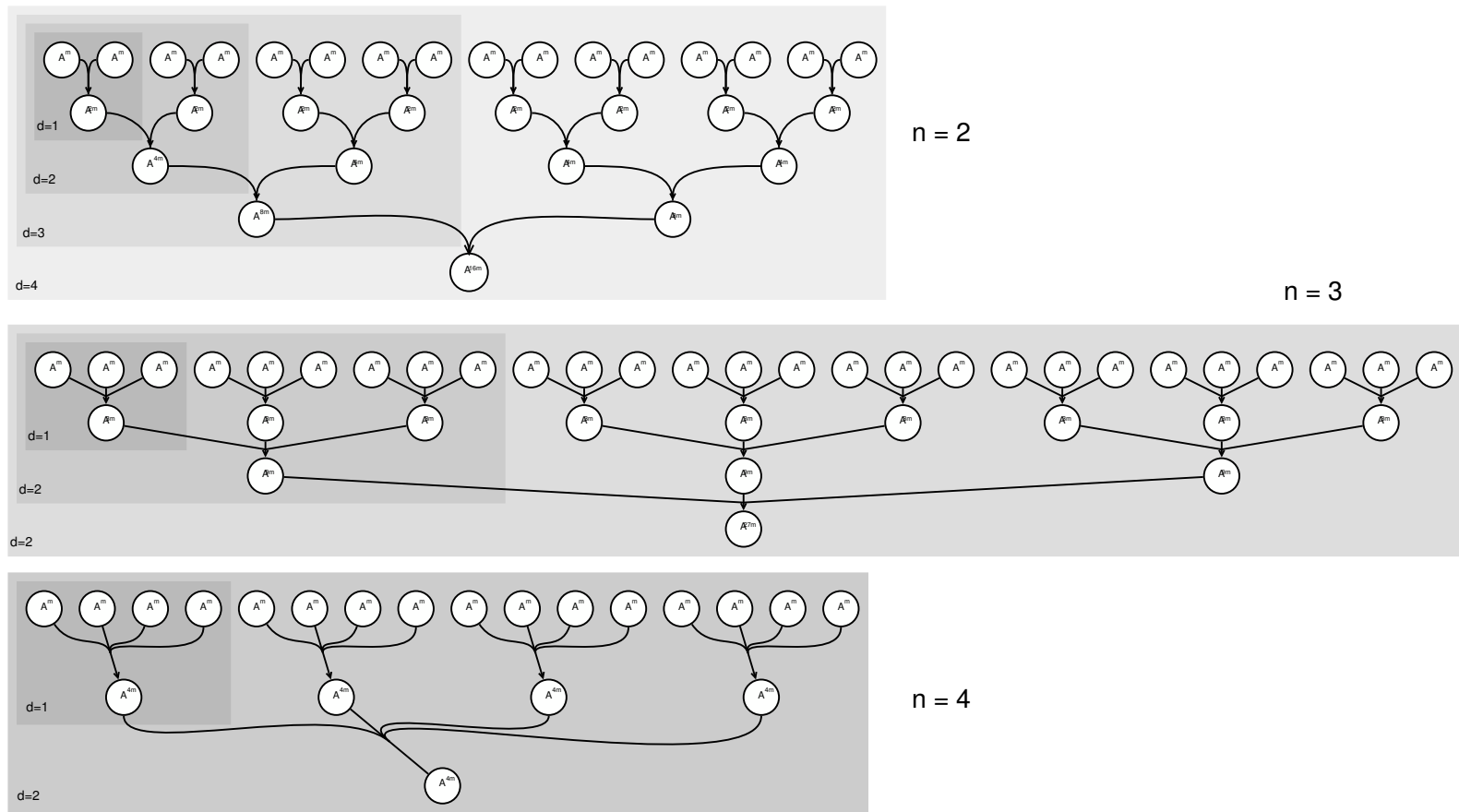


Figure 5.4: Exemplary reaction network layouts for a system where n identical chains are coupled for $n \in \{2, \dots, 4\}$. Note that for each depth-level (gray shaded box, d -level), the lowermost reaction correlates with the “removal from pool” reaction in Figure 5.5, top.

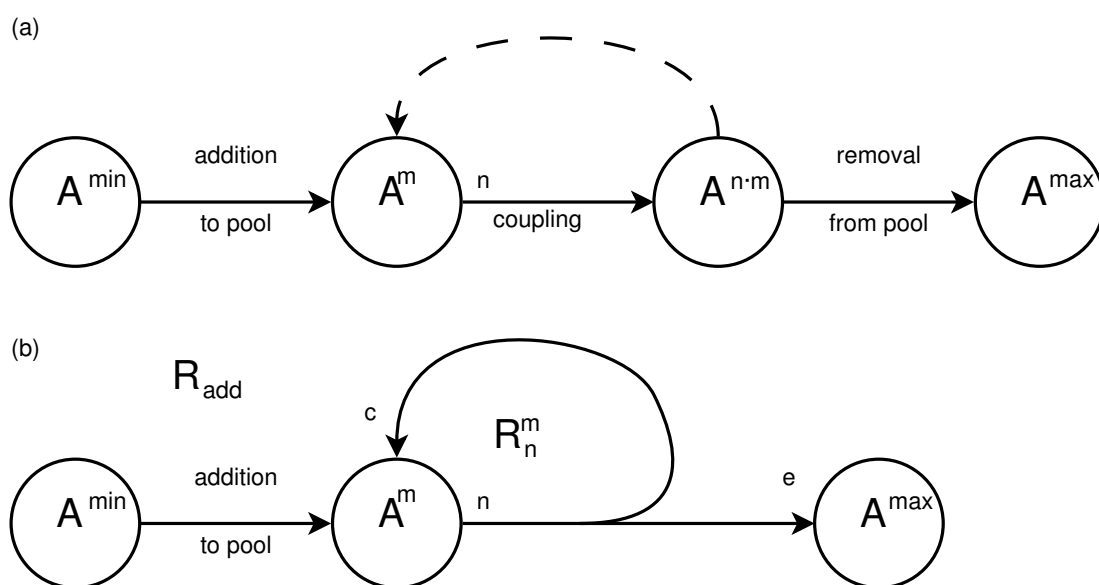


Figure 5.5: Building blocks for symmetric addition networks, where n identical chains of length m are coupled. Part (a) depicts a scheme for molecules entering a pool of polymers, their connection and the removal of polymers that have reached the maximum length. Part (b) shows a condensed, parametric representation of this loop, whose stoichiometric values c , e and n are discussed within Section 5.4.2

5. RULE-BASED METABOLIC DATABASES

Here, k represents the ratio between input and output polymer length or, in other words, the number of d-levels and is defined by

$$k = \log_n(max - min + 1).$$

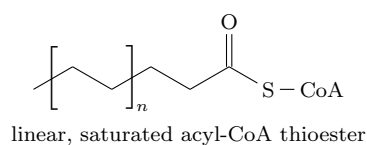
When forming the stoichiometric matrix for the reaction system, the contribution of this kind of reaction is

$$\mathcal{M} = \begin{matrix} & \dots & R_{add} & R_n^m & \dots & & \dots & R_{add} & R_n^m & \dots \\ \begin{matrix} \dots \\ A^{min} \\ A^n \\ A^{max} \\ \dots \end{matrix} & \begin{pmatrix} \dots & \dots & \dots & \dots \\ \dots & -1 & 0 & \dots \\ \dots & +1 & c-n & \dots \\ \dots & 0 & e & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} & = & \begin{matrix} \dots \\ A^{min} \\ A^n \\ A^{max} \\ \dots \end{matrix} & \begin{pmatrix} \dots & \dots & \dots & \dots \\ \dots & -1 & 0 & \dots \\ \dots & +1 & \frac{(1-n)n^k}{n^k-1} & \dots \\ \dots & 0 & \frac{n-1}{n^k-1} & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \end{matrix}$$

5.4.3 Example 3: Fatty Acid Degradation

A real world example for polymers of variable length are the naturally occurring acyl-CoA thioesters formed during fatty acid degradation: They are derivatives of those carboxylic acids mentioned in 5.1(1) and have a common substructure shown in Figure 5.6.

5.4.3.1 The Reaction System



chain length n	name	KEGG entry
0	Acetyl-CoA	C00024
1	Butanoyl-CoA	C00136
2	Hexanoyl-CoA	C05270
3	Octanoyl-CoA	C01944
4	Decanoyl-CoA	C5274
5	Dodecanoyl-CoA	C01832
6	Tetradecanoyl-CoA	C02593
7	Palmitoyl-CoA	C00154

Figure 5.6: Basic structure of linear saturated acyl-CoA thioesters and corresponding names for length up to 7

During one cycle of fatty acid degradation, these thioesters are oxidized to form dehydroacyl-CoA thioesters. These are then hydrated to hydroxyacyl-CoA thioesters, oxidized to oxoacyl-CoA thioesters, and finally an acetyl group is transferred to a

5.4 Homopolymers - Polymers Of Variable Length

- (A) acyl-CoA thioester : $CC\{-\}C\{+n\}CCC(=O)S\{CoA\}$
 (B) dehydroacyl-CoA thioester : $CC\{-\}C\{+n\}C=CC(=O)S\{CoA\}$
 (C) hydroxyacyl-CoA thioester : $CC\{-\}C\{+n\}C\{O\}CC(=O)S\{CoA\}$
 (D) oxoacyl-CoA thioester : $CC\{-\}C\{+n\}C\{=O\}CC(=O)S\{CoA\}$

1

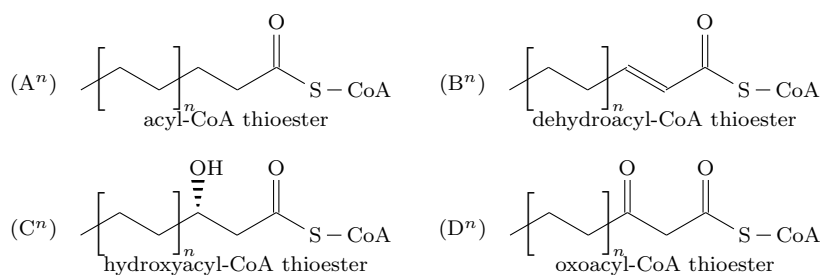
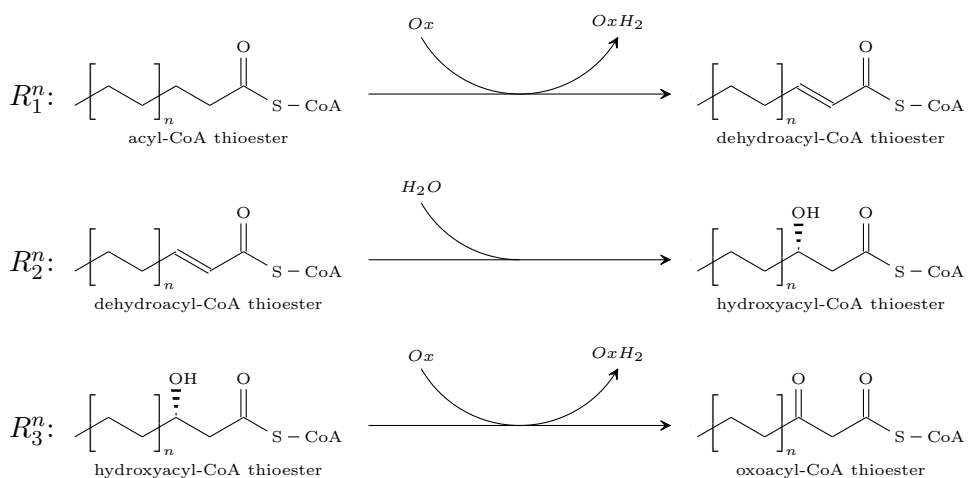


Figure 5.7: Main molecule classes involved in the fatty acid degradation cycle: CurlySMILES representation and fischer projection

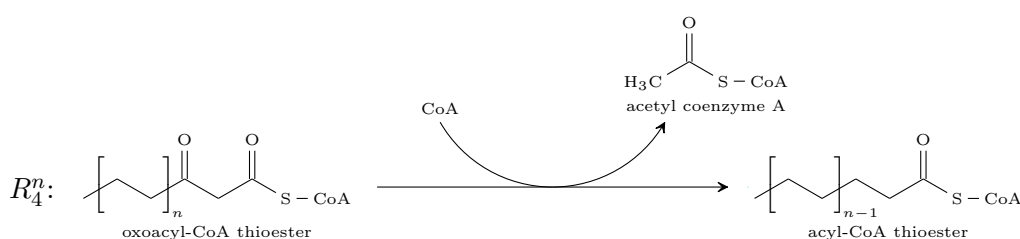
molecule of CoA. A new acyl-CoA thioester with a chain length reduced by one remains. An overview on these molecules and their CurlySMILES representation is given in Figure 5.7. For an initial length of $n = 7$, we have to perform 7 cycles, each detaching one molecule of acetyl-CoA, until no further shortening is possible. According to KEGG, we have at least four different reactions, each using a another acceptor, for the first step of each cycle. Also, for some of the other intermediate steps, several reactions are annotated in KEGG. Altogether, this sums up to a total amount of 45 different reactions acting on 42 chemical species. Using a rule based approach, it is possible to subsume the reactions to the following reaction types:



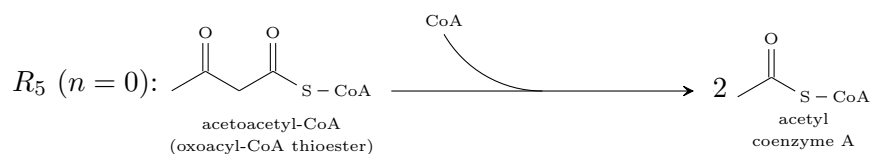
5. RULE-BASED METABOLIC DATABASES

Reaction	Possible oxidizing agents Ox for	
	$n = 0$	$n \in \{1 \dots 6\}$
R_1^n	ETF, FAD, O_2 , NAD^+ , $NADP^+$	ETF, FAD, O_2 , $NADP^+$
R_3^n	NAD^+ , $NADP^+$	NAD^+

Table 5.1: Possible acceptors for the different oxidation reactions within the fatty acid degradation



Although the majority of reactions is captured by this 4 generic reaction classes, there is one special step that has to be defined separately: The final cleavage of acetoacetyl-CoA to $2 \times$ coenzyme A.

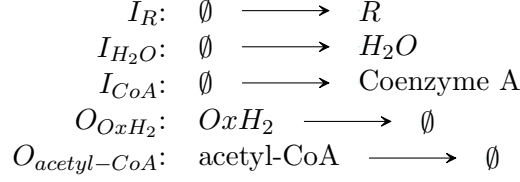


Note that the possible instances of the generic acceptor Ox may depend both on the length n of the polymer and on the reaction. Table 5.1 depicts the possible acceptor assignments as given by the KEGG database. The CLOSURE calculation algorithms presented in the interval chemistry section (5.4.1) can be applied for this network. However, there is even more that can be done:

5.4.3.2 Elementary Flux Mode Analysis

Basic calculations can be performed, using the stoichiometric matrix of the reaction system. For flux mode analysis we need to define inflow and outflow reactions for the external reagents:

5.4 Homopolymers - Polymers Of Variable Length



Now, the stoichiometric matrix can be set up:

$$\mathcal{M} = \begin{array}{l} Ox \\ OxH_2 \\ H_2O \\ CoA \\ acetyl - CoA \\ A^* \\ B^* \\ C^* \\ D^* \end{array} \begin{pmatrix} I_{Ox} & O_{OxH_2} & I_{H_2O} & I_{CoA} & O_{acetyl-CoA} & R_1 & R_2 & R_3 & R_4 & R_5 \\ +1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & +1 & 0 & +1 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & +1 & +2 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & -1 \end{pmatrix}$$

Note that A^* , B^* , C^* and D^* in the lower part of the left hand matrix legend represent the pools of all A^n , B^n , C^n and D^n molecules, respectively, and not n -specific instances. It is also possible to apply the algorithm for the determination of elementary flux modes proposed by Schuster et. al. (60). To demonstrate this approach, we extend the network described before by three reactions:

- R_6 : the (3R)-3-hydroxybutanoyl-CoA hydro-lyase reaction (KEGG entry R03027), which hydrates Crotonyl-CoA ($=B^0$) to form (R)-3-Hydroxybutanoyl-CoA
- R_7 : the (S)-3-Hydroxybutanoyl-CoA 3-epimerase reaction (KEGG entry R03276), which converts between the (R) and (S) epimers of Hydroxybutanoyl-CoA.
- R_8 : the (S)-3-hydroxybutanoyl-CoA hydro-lyase reaction (KEGG entry R03026), catalysing the hydratation of Crotonyl-CoA

Figure 5.8 gives an overview on the structure of the whole reaction system. In the first step of the algorithm, external metabolites are dropped and necessary sequential reactions are subsumed (Figure 5.9). Then the initial tableau is formed by transposing the stoichiometric matrix and augmenting it with the identity matrix:

5. RULE-BASED METABOLIC DATABASES

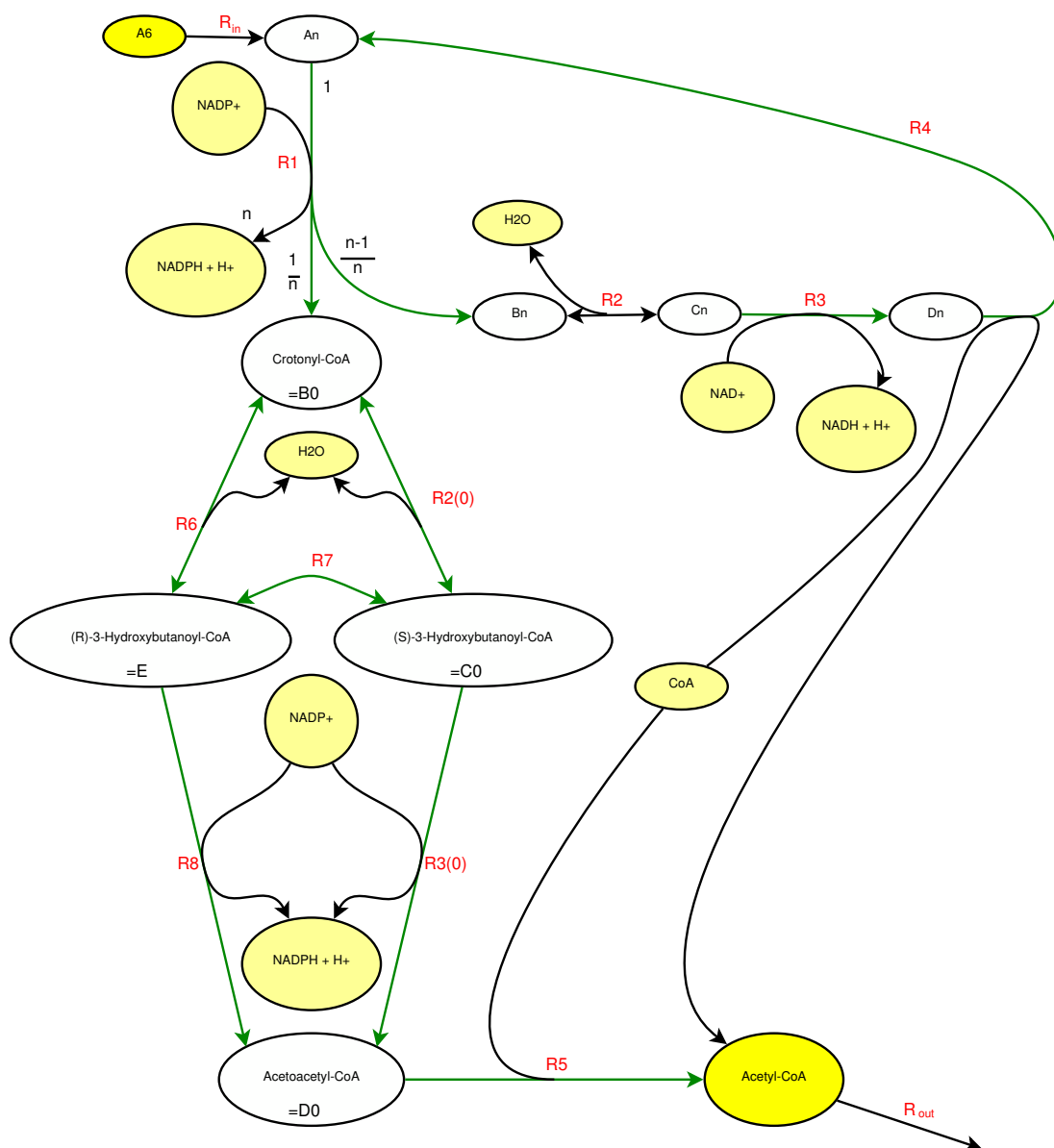


Figure 5.8: Overall reaction scheme for the biodegradation of fatty acids. Yellow bubbles indicate external species. Note that reactions R_2 and R_3 appear in two different forms: unbound (without specification of n in the upper part and bound ($n = 0$) in the central and lower part. Similarly, unbound instances of B, C and D are displayed.

5.4 Homopolymers - Polymers Of Variable Length

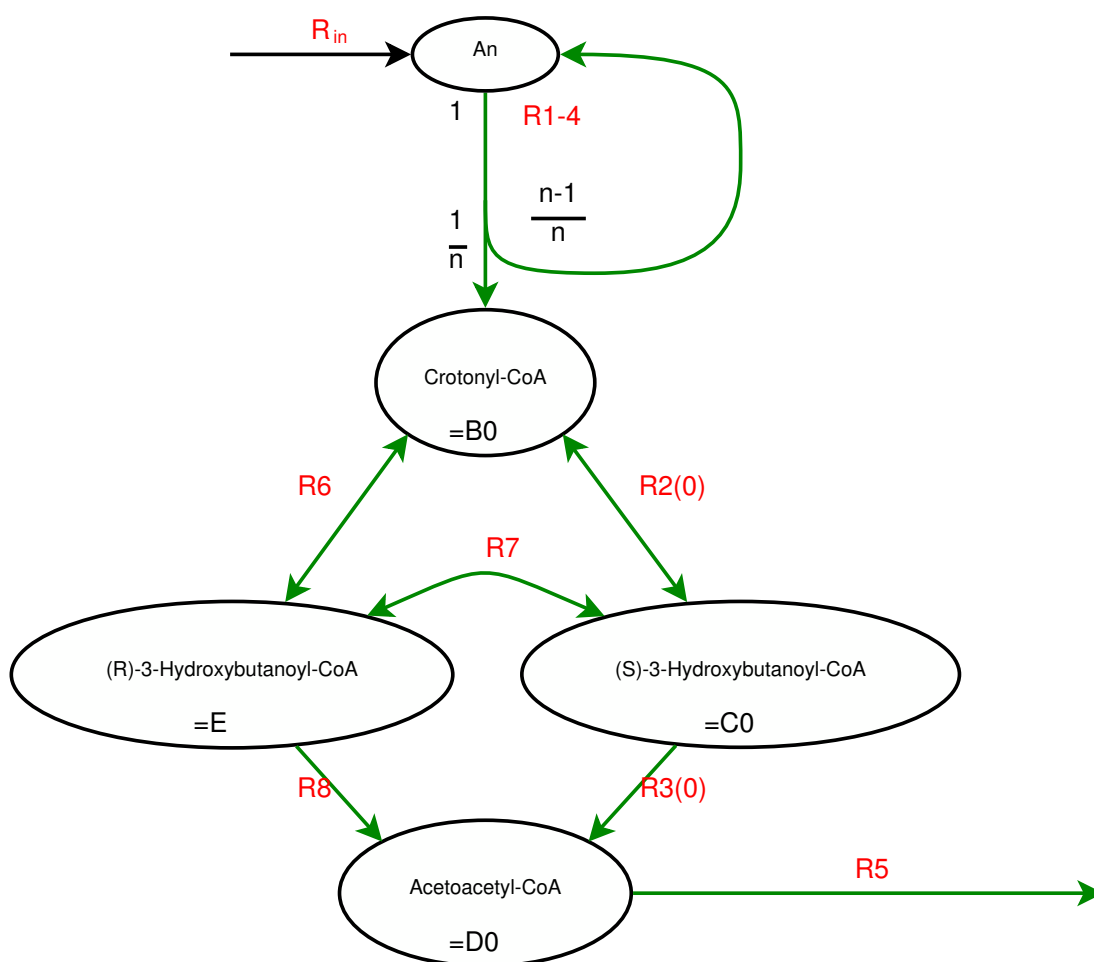


Figure 5.9: Reduced reaction scheme. Here the former reactions $R_1 \dots R_4$ are lumped together in R_{1-4} .

5.4 Homopolymers - Polymers Of Variable Length

binations:

$$\mathcal{T}^3 = \begin{array}{c} R_8 \\ R_5 \\ \mathcal{T}^1[2] + \mathcal{T}^1[7] \\ \mathcal{T}^2[1] + \mathcal{T}^2[3] \\ \mathcal{T}^2[5] - \mathcal{T}^2[1] \\ \mathcal{T}^2[3] + \mathcal{T}^2[5] \end{array} \begin{array}{c} A_n \quad B_0 \quad C_0 \quad D_0 \quad E \\ \left(\begin{array}{cccc|cccccccc} 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 \\ 0 & 0 & 0 & 0 & +1 & 0 & +1 & 0 & +1 & +n & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & -1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & +1 & +1 & 0 & -1 & +1 & +n & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & +1 & 0 & 0 & +1 & +n & 0 & +1 & 0 & 0 \end{array} \right) \end{array}$$

$$\mathcal{T}^4 = \begin{array}{c} \mathcal{T}^1[2] + \mathcal{T}^1[7] \\ \mathcal{T}^2[5] - \mathcal{T}^2[1] \\ \mathcal{T}^3[1] + \mathcal{T}^3[2] \\ \mathcal{T}^3[2] + \mathcal{T}^3[4] \\ \mathcal{T}^3[2] + \mathcal{T}^3[6] \end{array} \begin{array}{c} A_n \quad B_0 \quad C_0 \quad D_0 \quad E \\ \left(\begin{array}{cccc|cccccccc} 0 & 0 & 0 & 0 & +1 & 0 & +1 & 0 & +1 & +n & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & +1 & 0 & -1 & +1 & +n & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & +1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & +1 & +1 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & +1 & +n & 0 & +1 & +1 & +1 \end{array} \right) \end{array}$$

The final tableau is

$$\mathcal{T}^5 = \begin{array}{c} \mathcal{T}^3[2] + \mathcal{T}^3[6] \\ \mathcal{T}^4[1] + \mathcal{T}^4[3] \\ \mathcal{T}^4[1] + \mathcal{T}^4[4] \end{array} \begin{array}{c} A_n \quad B_0 \quad C_0 \quad D_0 \quad E \\ \left(\begin{array}{cccc|cccccccc} 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & +1 & +n & 0 & +1 & +1 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & +1 & +n & +1 & 0 & +1 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 & +1 & +n & 0 & +1 & +1 & +1 \end{array} \right) \end{array}$$

From this tableau, the following elementary modes (of the condensed network) are derived (reactions ordered by appearance in flux):

- $\{R_{in}, n \times R_{1-4}, R_2(0), R_3(0), R_5\}$
- $\{R_{in}, n \times R_{1-4}, R_6, R_8, R_5\}$
- $\{R_{in}, n \times R_{1-4}, R_6, R_7, R_3(0), R_5\}$

They can be expanded (cf. Figure 5.10) to:

- $\{R_{in}, n \times R_1, n \times R_2, n \times R_3, n \times R_4, R_2(0), R_3(0), R_5\}$
- $\{R_{in}, n \times R_1, n \times R_2, n \times R_3, n \times R_4, R_6, R_8, R_5\}$
- $\{R_{in}, n \times R_1, n \times R_2, n \times R_3, n \times R_4, R_6, R_7, R_3(0), R_5\}$

In a similar fashion, the algorithm for calculating elementary modes may be applied to all kinds of rule-based networks.

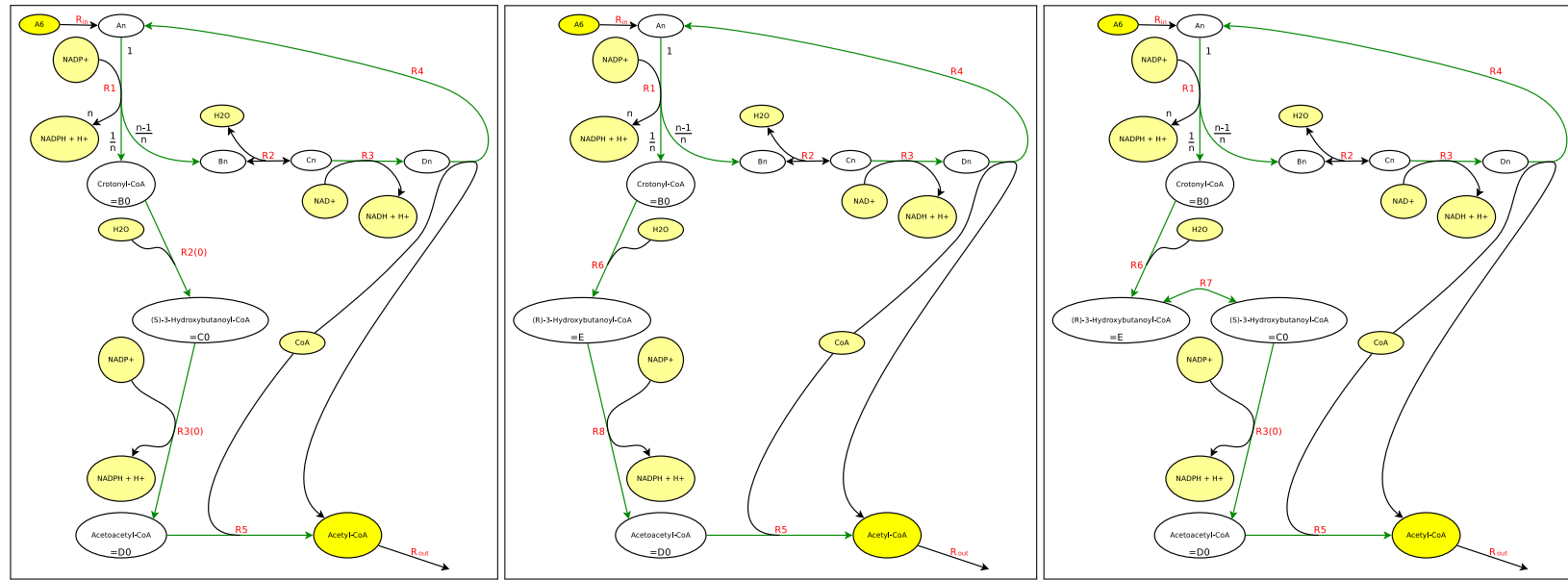


Figure 5.10: The three elementary modes found by the algorithm. Note that the mode $A_n \Rightarrow B_0 \Rightarrow C_0 \Rightarrow E \Rightarrow D_0$ is also elementary, but may be represented as a linear combination of the other modes.

5.5 Linear Polymers With Several Building Blocks

The manual investigation of a quarter of our database excerpt from KEGG gave an estimate of about 10 classes of naturally occurring linear polymers with multiple building blocks. This may, however, be a misleading result, as many substances building hetero-polymeric structures may not be annotated as polymers and thus are not found by our database search. The results found within the data set almost completely either belong to the class of nucleic acids or to the class of peptides. Apart from this, there are a few types of sugars molecules which form polymers of two alternating base units, while most polyglycans tend to form branched polymers. Well-known examples of nucleic acids are DNA, RNA and more specialized forms like mRNA and tRNA. Similarly, polymers like proteins or glycopeptides are well-known representatives of the class of polypeptides. The aforementioned examples can be divided into two further partitions:

1. linear heteropolymers with a fixed ratio of alternating building blocks and
2. polymers that vary in their module sequence.

5.5.1 Fixed Ratio Heteropolymers

A prominent example of a fixed ratio heteropolymer is agarose, consisting of alternating units of D-galactose and L-anhydrogalactose (KEGG Id C01399). As long as the modules occur in the same sequence of each repetition, those repetitive subsequences can be treated as the ultimate building blocks of a linear homopolymer (cf. Section 5.4).

5.5.2 Aperiodic Linear Polymers

Polymers without a fixed sequence of monomeric units, however, are much more intricate. Still, for many applications, like balancing chemical reaction networks, it may be sufficient to reduce the complexity to the amounts of each basic unit. An example: as a polynucleotide with 10 bases can have $4^{10} \approx 1,000,000$ different sequences, it is easier to work with the amounts $\#A, \#C, \#G, \#T$, with $\sum \#A + \#C + \#G + \#T = 10$ (which are limited to $\sum_{i=0}^{10} \sum_{j=0}^i \sum_{k=0}^j = 286$ possibilities). This simplification would be sufficient for any approach where one is just interested in the amounts of atoms, that is, in cases where the exact sequence of nucleotides does not matter. If, however, for

5. RULE-BASED METABOLIC DATABASES

any reason the block sequence in a polymer is important and needs to be distinguished, polymers with the same ratio of modules but in different order have to be distinguished and cannot be subsumed into equivalence classes. Even for those polymers whose sequence is important, it may be possible to use the equivalence class approach which will be explained by a simplified model of the DNA creation process (cf. Section 4.5.2.2):

On the one hand, when a cell produces DNA, it aims for a very specific sequence of that molecule rather than a random sequence. This means that the order in which the triphosphonucleotides are added is fixed and the whole molecule can be written as the parametrized molecule DNA(S), with S being the sequence string.

On the other hand, the attachment of activated nucleotides is guided by a complementary strand, meaning that if a functional DNA replication apparatus is present, one can assume that the order in which the bases are added to the system does not play a major role, the DNA-Polymerases “taking care” of this order. In general, in the presence of functional replication systems we only have to create a DNA molecule with the right amounts of each base – and will only gain molecules in the right composition.

Therefore, for modelling purposes it is sufficient to model this polymer as parametrized by the amounts of the four bases: DNA(a,c,g,t), with $a, c, g, t \in \mathbb{N}$. Then, the same transition pattern as in Section 5.4.2 can be applied, where all adenines (A), cytosines (C), guanines (G) and thymines (T) are added in independent cycles, each acting on a molecule with one free parameter. In the following, we describe the process of the creation of a DNA strand with the sequence GATTACA¹. This sequentially ordered process is illustrated in Figure 5.11a, the reaction scheme described in the following is depicted in Figure 5.11b.

1. The starting point is an “empty” DNA molecule, DNA(0,0,0,0)
2. This “empty” DNA is then introduced into a pool DNA(a,0,0,0) of DNA molecules with a variable number of adenines, and zero cytosines, guanines and thymines.
3. To this pool of molecules – unbound in their amount of adenines, but with fixed (zero) amounts of C, G and T – the right amount of adenines is added. This addition is reflected by a cyclic reaction as described in Section 5.4.2. The branching coefficients are $\frac{1}{n}$ and $\frac{n-1}{n}$ with $n = 3$ being the number of As which are present in the target molecule.

¹<http://www.imdb.com/title/tt0119177/>

5.5 Linear Polymers With Several Building Blocks

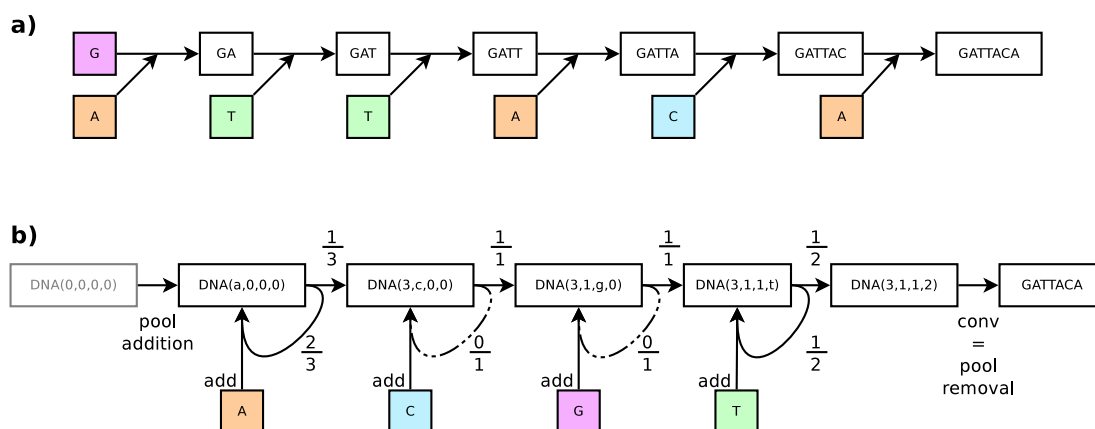


Figure 5.11: Schematic reaction path for creation of a DNA with sequence “GATTACA”.
¹ Subfigure (a) displays the reactions in the order of base attachment. Part (b) simplifies this scheme by the cyclic application of four steps corresponding with the amount of the four bases in the final DNA.

4. In the next step² in a (non-circulating) cycle the only cytosine is added.
5. In the same way, the sole molecule of guanine is appended to the polymer.
6. Afterwards, in another performing cycle, two thymines are added.
7. Finally, from the pool of DNA molecules, with 3 As, 1 C, 1 G, and 2 Ts, the molecules with the right sequence, GATTACA, are picked as outflow. Again, these molecules are the only ones which will be in this pool, as the overall elongation process is guided by a complementary DNA strand.

For this reaction system, we can construct the following stoichiometric matrix, where

²these steps have been sorted in alphabetical order and could be exchanged

5. RULE-BASED METABOLIC DATABASES

DNA(0,0,0,0) is disregarded:

$$\mathcal{M} = \begin{matrix} & A & C & G & T & D(a000) & D(3c00) & D(31g0) & D(311t) & D(3112) & GATTACA \\ \begin{matrix} R_{in}^A \\ R_{in}^C \\ R_{in}^G \\ R_{in}^T \\ add^0 \\ add^A \\ add^C \\ add^G \\ add^T \\ conv \\ out \end{matrix} & \left(\begin{array}{ccccccccccc} +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -\frac{1}{3} & +\frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -\frac{1}{2} & +\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{array} \right) \end{matrix}$$

Note that in the matrix DNA(a,0,0,0), DNA(3,c,0,0), DNA(3,1,g,0), DNA(3,1,1,t), and DNA(3,1,1,2) are abbreviated as D(a000), D(3c00), D(31g0), D(311t) and D(3112), respectively. Based on this matrix it is possible to calculate the sole elementary flux as $3 \times R_{in}^A + R_{in}^C + R_{in}^G + 2 \times R_{in}^T + add^0 + 3 \times add^A + add^C + add^G + 2 \times add^T + conv + out$, exactly mirroring the process. With this approach, it is possible to precisely calculate the chemicals used up by the DNA replication process in a molecular model of a cell with a known genome sequence.

5.6 Attachments And Charges

Attachments and charges can usually be handled similar to heteropolymers: Usually an attachment is an exchangeable chemical group covalently bound to a molecule part with a fixed structure. Actually, most of the building blocks forming the preceding heteropolymers are equal molecules with different side groups. For example, the exchangeable nucleotides used in nucleic acids are not five completely different molecules. They are rather similar molecules possessing the same basic structure: A (deoxy) ribose sugar, to which different nucleobases are attached. Accordingly, instead of describing the whole nucleotide with one of the letters A, C, G, T, or U, it would be possible, although cumbersome, to describe just the nucleobases with these letters and annotate a nucleotide as $C_5H_{10}O_4 - \mathbf{A}$, $C_5H_{10}O_4 - \mathbf{C}$, $C_5H_{10}O_4 - \mathbf{G}$, $C_5H_{10}O_4 - \mathbf{T}$, or $C_5H_{10}O_4 - \mathbf{U}$. Also, charges of ions can be represented as symbols or numbers. For example A^{n+} is a common representation for a positively charged ion, where n determines the actual charge.

5.7 Branched And Cross-linked Structures

This extends the previously mentioned classes of molecules beyond linearity. Within suchlike rule-based molecules, we have branches within the variable regions of our molecules, that can even form loops or meshes. Although the CurlySMILES framework provides means to annotate molecules with such structures, our presented formal description cannot capture these molecules: The formal definition introduced in this manuscript only allows numbers and strings as parameter types, which are linear representations of molecules whose variable regions are linear. Establishing a formalism for such molecules is beyond the scope of this paper, but certainly would be a very interesting topic for future work.

5.8 Discussion

Rule-based annotations may be the future of metabolic databases. The aforementioned formalism allows to define parametric molecule classes and rule-based reactions in a consistent, yet intuitive way. With the CurlySMILES annotation we already possess a well defined and powerful tool to describe linear parametric molecules. The drawback of this format is its incapability for parametric molecule chains with several building blocks and branched structures. The defined molecule and reaction classes lay the foundation for new possibilities and challenges on the algorithmic side of metabolic theory. Some simple algorithms for basic calculations on rule-based reaction systems have been shown, and it was described how some basic methods can be adapted to linear molecules. However, for branched and crosslinked structures, annotation formats still need to be developed before we even can think about algorithms. Also, being able to annotate metabolic networks and calculating fluxes and balances without expanding the metabolite classes to specific sets of molecules could offer new possibilities for high-throughput approaches.

5. RULE-BASED METABOLIC DATABASES

6

CONCLUSION AND OUTLOOK

When integrating large-scale models from various data sources, inconsistencies inevitably emerge. In this thesis it has been demonstrated how some of these inconsistencies can be detected and corrected automatically (Chapter 2, 4).

In particular, a method for the analysis of SBML models based on organization theory was developed. This method is able to reveal structural information hidden in kinetic laws of SBML embedded reactions (Chapter 2). The application of this algebraic analysis method to the Biomodels Database uncovered 31 models where reactions were missing in reactive organizations. Scanning the whole database and comparing our results to the results achieved by other methods like FBA and extreme pathway analysis confirmed the usefulness of our method.

After this, a procedure for the prediction of biodegradation pathways and its prerequisites was presented (Chapter 3). Based on the conditions prevailing within the targeted environment, the necessary steps to find degradation pathways for certain compounds and propose microbial communities with the ability to perform this degradation were described. These steps were designed in accordance with well known algorithms like breadth-first-search, EMA, FBA, and other variants of ILPs and aim to combine metabolic networks composed of multiple species. Moreover, it was shown how this multi-organism approach increases the demand for integrating data from various organisms and sources, like, for example, network models of specific organisms formulated in SBML or other appropriate languages or online metabolic databases. In principle, both the data integration task and the pathway search problem appear to be automatically solvable, while, however, a non-interactive algorithmic solution is

6. CONCLUSION AND OUTLOOK

hindered by annotational weaknesses, ambiguous data and incorrect entries in online databases.

Our focus on these inconsistent database entries motivated a closer look on one of the largest metabolic databases currently available: Therefore, an analysis of the KEGG database was performed, looking for inconsistencies in molecular formulas and reactions (Chapter 4). This revealed a variety of problems based on human-readable comments or semantic inconsistencies and thus hardly detectable by means of algorithms. Yet on the other hand there are lots of flaws which are indeed easily detectable, like ambiguous data, syntactically incorrect database entries, and missing information; most of them implying sets of reactions not balanced on the atomic level. Other problems with molecule and reaction entries, however, pose a more complicated task due to variables that are used without prior declaration. In many cases such reactions might be balanced only for certain values of the variables included. Additionally, there are database entries that miss some information regarding the chemical structure of molecules or reactions. Yet these entries, even though they are only partially balanced or lack information, can be detected algorithmically, too. A large portion of issues trace back to syntactic flaws and can be corrected automatically; finding missing atoms or groups by calculating balances based on well-known parts or by referring to balances of similar reactions. In contrast, reactions for cases where multiple reactants with unknown formula are included cannot be balanced, as well as those reactions which are only shorthands for multistep reaction paths.

In a further step, it has been revealed that inconsistency correction might prove insufficient. Thus, the underlying model descriptions will have to be re-formulated and improved. For some specific kinds of molecules and reactions this might be almost impossible, since the storage formats used do not account for polymeric or modular structures. To still fulfill the demands mentioned, new tools would be needed, allowing for the handling of aspects of data currently formulated informally in human-readable texts. All these problems demonstrate a strong need for extensive curation of semantic contents and for annotations that are both well defined and flexible. To achieve that, an innovative, simplified rule-based modeling approach has been suggested (Chapter 5). The advantage of this method, compared to more general approaches like BNGL (19), is its simplicity allowing for a simplified handling both by humans and computer algorithms. Despite its simplicity, the new approach enables us to represent the combinatorial information gathered in current databases like KEGG (Chapter 4), which is also demonstrated by formulating nucleotide activation and DNA elongation using our rule-based formalism.

Consequently, the implementation of these rule-based databases is a challenging task and lays the foundation for more precise annotations and new analytic algorithms. The data integration and combinatorics achieved with such rule-based data may provide new insights into biochemical networks and contribute to an interesting field of research in the realm of computational biology.

6. CONCLUSION AND OUTLOOK

7

SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

Christoph Kaleta^{*,1,2}, Stephan Richter¹, and Peter Dittrich²

7.1 Emulating Flux-based Network Analysis Methods

Flux-based methods allow to predict whether a reaction can appear in any steady state. These methods have additional requirements on the network, like definition of upper bounds of inflow reactions for FBA (65) or network size in elementary mode analysis (59) and extreme pathway analysis (58). Since we only want to test whether a reaction can appear in a steady-state flux a simplified approach that yields the same results like elementary mode analysis as well as extreme pathway analysis can be used. This approach is similar to FBA and allows to determine whether a reaction can appear in any steady state of the model using linear programming.

Given the stoichiometric matrix \mathbf{N} of a model of n reactions for which we want to determine whether reaction i can appear in a steady-state flux, the constraints of the linear program are

$$(1) \mathbf{N} \cdot \mathbf{v} = 0$$

¹Authors contributed equally

²to whom correspondence should be addressed (dittrich@minet.uni-jena.de)

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

$$(2) \mathbf{v} \geq 0$$

$$(3) v_i \geq 1$$

with \mathbf{v} being the vector of variables. A dummy objective function can be used if required. Since we have not defined any upper bound, constraint 3 is equal to searching a steady-state flux \mathbf{v} with a positive flux in v_i . Since we are also interested in states where some species can accumulate over time, the steady-state condition is relaxed to a constraint similar to the self-maintenance condition in chemical organization theory (OT). Thus, when checking if reaction i can be part of a flux of the reactions such that the concentration of no molecule declines, the constraints read

$$(1) \mathbf{N} \cdot \mathbf{v} \geq 0$$

$$(2) \mathbf{v} \geq 0$$

$$(3) v_i \geq 1$$

If the linear program is feasible, flux-based methods would predict i to be present in a flux of the system where the concentration of no molecule declines.

7.2 Extending The Steady-state Lemma To Growth States

In Dittrich2007 (17) it is shown that each steady state of a reaction network can be mapped to an organization of the system, if the reaction network obeys the condition that the kinetic law of a reaction implicates a non-zero flux if and only if all educts have a positive concentration (20). Additionally we assume that during simulation no species has a negative concentration and each flux is positive, i.e., reversible reactions are split into irreversible forward and backward reactions. Here we will demonstrate that, if the network is simulated using ordinary differential equations, the steady-state lemma can be extended to every phase of the simulation in which there is a non-negative concentration change for each species.

Given two points $\mathbf{x}(t_1)$ and $\mathbf{x}(t_2)$ with $0 < t_1 < t_2$ and $\mathbf{x}(t_1), \mathbf{x}(t_2) \in \mathbb{R}^n$ in the trajectory $\mathbf{x}(t)$ of the concentration of the species of a reaction network during simulation, we call the time span $[t_1, t_2]$ a *growth phase* if $\mathbf{x}(t_1) \geq \mathbf{x}(t_2)$. As the species set s_t present at a time-point t we identify each species having a positive concentration, i.e., $s_t = \{i \mid x_i(t) > 0\}$. If the kinetic laws of the network fulfill the Feinberg conditions s_t necessarily fulfills the closure-condition if $t > 0$. If we determine s_t at a time-point

7.2 Extending The Steady-state Lemma To Growth States

t_g during a growth phase, $t_g \in [t_1, t_2]$, there additionally exists a flux vector d fulfilling the self-maintenance condition for s_{t_g}) and hence s_{t_g} constitutes an organization. The existence of d will be demonstrated in the following. In passing we note that a non-negative concentration change for each species in the interval $[t_1, t_2]$ implies that $s_{t_g} = s_{t_1} = s_{t_2}$.

Given the stoichiometric matrix \mathbf{N} of a reaction network and the kinetic laws of the reactions as $\mathbf{v}(t)$, the ordinary differential equation

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{N}\mathbf{v}(t) \quad (7.1)$$

describes the trajectory of the concentrations of the species $\mathbf{x}(t)$ given the starting point $\mathbf{x}(0)$. Usually $\mathbf{v}(t)$ is written as being dependent on $\mathbf{x}(t)$. Here we assume that we have already solved the differential equation in (7.1) since the following proof relies only on the existence of the solution $\mathbf{v}(t)$. Thus, we can compute the concentration change of the species of the network in the interval $[t_1, t_2]$ as

$$\mathbf{x}(t_2) - \mathbf{x}(t_1) = \int_{t_1}^{t_2} \frac{d\mathbf{x}(t)}{dt} dt = \int_{t_1}^{t_2} \mathbf{N}\mathbf{v}(t) dt = \mathbf{N} \int_{t_1}^{t_2} \mathbf{v}(t) dt \quad (7.2)$$

Now, if we assume that the system is in a growth phase during t_1 and t_2 we can choose \mathbf{d} as

$$\mathbf{d} = \int_{t_1}^{t_2} \mathbf{v}(t) dt \quad (7.3)$$

Since we assume $\mathbf{v}(t)$ to be non-negative, \mathbf{d} has only non-negative entries. From the condition, that every reaction has a non-zero flux, if and only if all its substrates are present, i.e., have a positive concentration, we can derive that \mathbf{d} has positive values for each reaction implied by the species set s_{t_g} . Additionally we can see from (7.2) that $\mathbf{N}\mathbf{d} \geq 0$, hence, \mathbf{d} fulfills the self-maintenance condition for s_{t_g} . This implies, that s_{t_g} is an organization.

In consequence, each growth phase of the simulation of a reaction network corresponds to an organization of the system. Moreover, if there exists a *growth state* of the network, i.e., $t_2 \rightarrow \infty$ this state also corresponds to an organization. Please note that a growth state contains the steady-state condition, i.e., $\mathbf{N}\mathbf{v} = 0$ and $\mathbf{v} \geq 0$, as special case. Hence, the steady-state lemma can be generalized to a growth-state lemma.

7.3 Detailed Description Of The SBML Processing Algorithm

7.3.1 Libraries Used

We implemented our analysis tool in Java and used the JigCell SBML parser, available under the DARPA BioCOMP Open Source License on <http://jigcell.biol.vt.edu>. This code is used to open, modify, and save the analyzed SBML models.

7.3.2 Overview On The Processing Steps

In order to perform the OT analysis, the model is passed through several analysis and adjustment steps:

1. reading and testing the SBML code
2. searching for defined meta-ids
3. building a look-up table for used functions
4. building a look-up table for predefined parameters

For each reaction in a model, we perform:

5. analysis of the structure of the kinetic laws
6. adaptation of the reaction structure to fulfill the Feinberg condition
7. adaptation of the kinetic laws to preserve the dynamics

7.3.3 Description Of The Steps

In the subsequent sections we will use the following terms:

- *support*: a modifier set is supporting a reaction if a non-zero concentration of the modifiers allows a non-zero flux of the reaction
- *absent, absence, deletion*: a modifier's concentration is set to zero

7.3.3.1 Reading And Testing The SBML Code

Prior to all analysis steps the models have to be read in. This is done using the JigCell SBML parser, which also checks the syntactic structure while loading the document. As a result of this syntax checking, we found *MODEL8262229752* to contain syntactical errors.

7.3.3.2 Searching For Defined Meta-ids

All reaction and species entities in a model have a unique id. For the later creation of new reactions it is necessary to create new ids, which is only possible, if we know the existing ids in the model. New ids are given names like `metaid_XXXXXXX`, where `XXXXXXX` stands for the first free number including leading zeros.

7.3.3.3 Building Function And Parameter Look-up Tables

For several subsequent steps, we need a table of all used function names, their respective parameters and the assigned function. Therefore, a data structure mapping each function name to this information is created by analyzing the model's function definitions. In a similar way, all kinetic parameters defined in the SBML model are stored in a mapping structure, which allows a replacement of parameter occurrences by their values when resolving kinetic laws.

7.3.3.4 Analysis Of The Structure Of The Kinetic Laws

The most important and complex step in the process is the examination of the kinetic law of each reaction. For this purpose, every rate law in the SBML document is parsed into a tree structure using the JigCell library. The tree structure of reaction 3 of the example network is shown in Figure 7.1. The main goal of this step is to gather

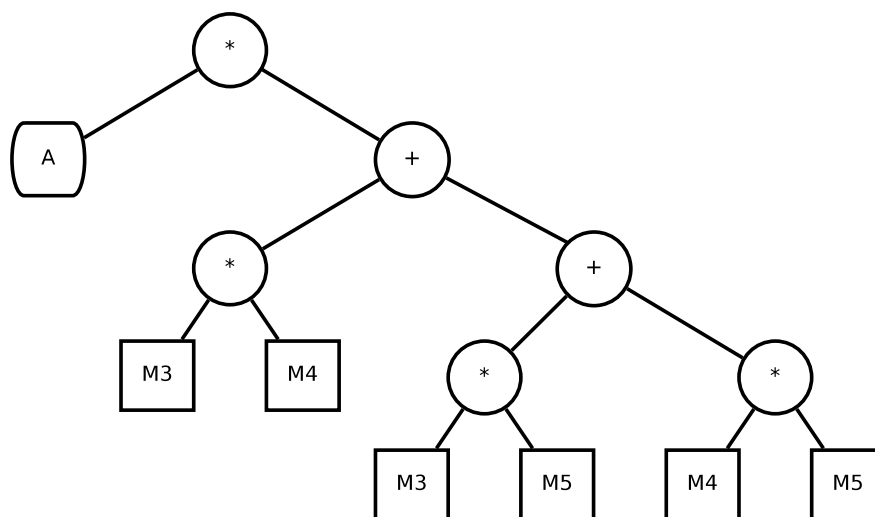


Figure 7.1: The tree corresponding to the rate law of reaction 3 from the example network given in the main paper. Circles indicate operator nodes, round boxes denote species and squared boxes correspond to parameters.

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

the set of all modifiers involved in a reaction and, moreover, to determine the set of supporting sets. This is achieved by checking which modifiers are omittable, i.e., can be absent without reducing the flux of the reaction to zero. Therefore, a list of all modifiers is obtained from the reaction definition. Out of this set, the power set of all involved modifiers is calculated and passed to a data structure, which we will refer to as *untested modifiers* in the following. Please note that if a set of modifiers is replaced by zero values, this means the concentrations of the modifiers in the complement set are left positive. Therefore, we check the support of a single modifier by testing the effect of deleting all other modifiers. The following steps are performed iteratively over all sets in the untested set:

The largest untested set is obtained from the data structure. The modifier species contained within this set are assumed to be absent. Thus, all their occurrences are replaced by zero values in the kinetic law. Literally spoken, we test support of the smallest set by deleting the largest. In our example, the first untested set would be $\{M3, M4, M5\}$. Hence, these modifiers are replaced by zero values, as can be seen in Figure 7.2. Then, beginning from the leaves, the tree structure (related to the

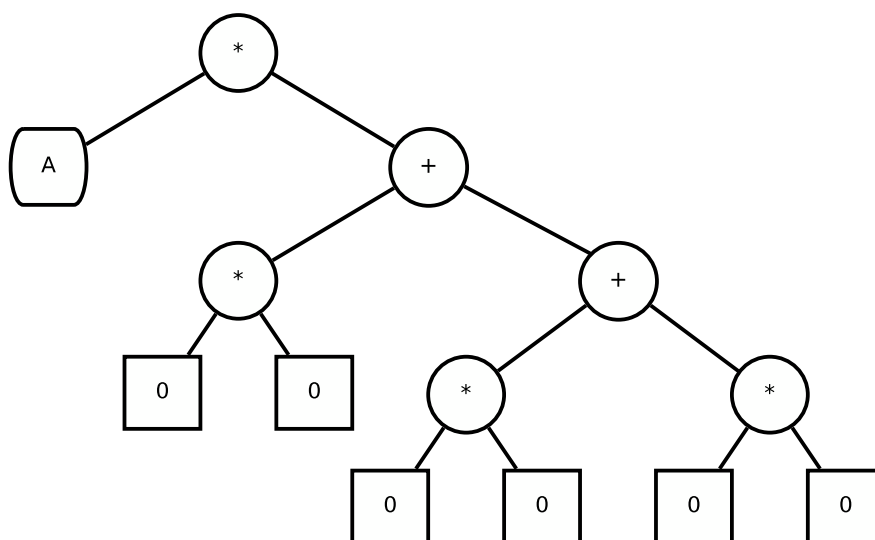


Figure 7.2: The same tree after application of zero values to the (absent) modifiers.

current kinetic law) is resolved by application of mathematical rules. Simultaneously, all occurrences of parameters are replaced by their values from the look-up table. In the example, the right subtree of the root node will be solved to zero, while the left leaf node will be replaced by the value of A , if given, or stay a symbol otherwise (Figure 7.3). As one can see, if all modifiers are absent, this reaction will have a zero flux. Consequently,

7.3 Detailed Description Of The SBML Processing Algorithm

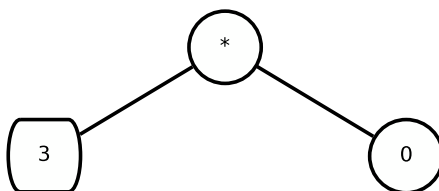


Figure 7.3: The right subtree solved to zero, the left subtree was replaced by A 's value, say 3

the set $\{M3, M4, M5\}$ is marked as not omittable and removed from the untested sets. Due to the ordering by size, the next sets to be tested are $\{M3, M4\}$, $\{M3, M5\}$ and $\{M4, M5\}$. In each case we find that the flux in the reaction is constrained to zero if the modifiers are deleted (Figure 7.4). Informally spoken, this means: if you delete

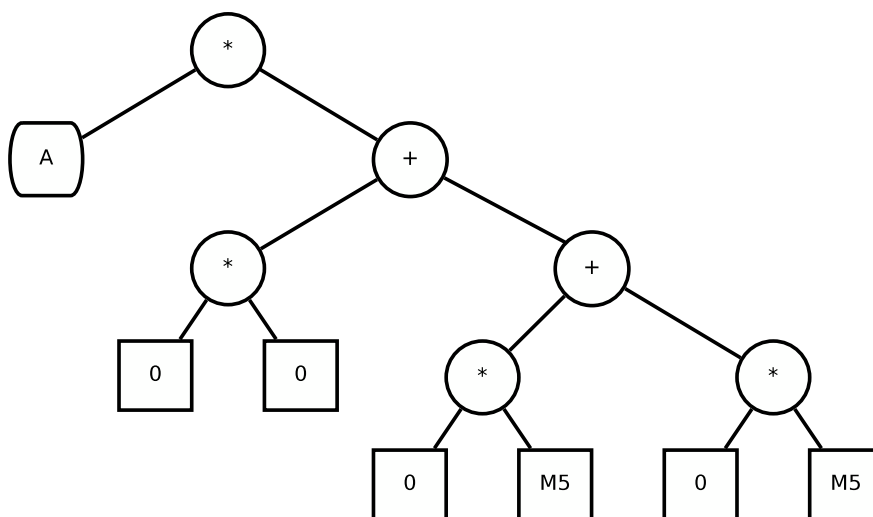


Figure 7.4: The right subtree is solved to zero, if the two modifiers $M3$ and $M4$ are absent

any pair of modifiers in $R3$, the reaction will stop, or, no single modifier supports the reaction. Therefore those sets are marked as not omittable, too.

The next untested sets are $\{M3\}$, $\{M4\}$ and $\{M5\}$. As can be seen by application of a zero concentration to modifier $\{M3\}$, we no longer obtain a zero value as result, but a function depending on the concentration of $\{M4\}$ and $\{M5\}$. As we assume their concentrations to be positive, we find that reaction $R3$ can have a positive flux if $M3$ has a zero concentration (Figure 7.5). The single-modifier sets $\{M3\}$, $\{M4\}$ and $\{M5\}$ are marked as omittable and hence their complements $\{M4, M5\}$, $\{M3, M5\}$ and $\{M3, M4\}$ are the supporting sets of the reaction. Since those sets already enable

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

a positive flux in the reaction, there is no need to additionally test whether the entire modifier set is a supporting set. In our algorithm, this is achieved by marking all proper subsets of the sets $\{M3\}$, $\{M4\}$ and $\{M5\}$ as omittable, which accordingly means, that all supersets of $\{M4, M5\}$, $\{M3, M5\}$ and $\{M3, M4\}$ are recognized to support the reaction.

7.3.3.5 Adapting The Reaction Structure And The Kinetic Laws

After determining the minimal supporting modifier sets, we adapt the reactions in order to correctly apply OT. Our algorithm distinguishes 3 cases:

1. None of the modifiers is omittable, i.e., the (only) supporting set is the entire modifier set. In this case all modifiers are removed from the reaction and added as educts and products to the reaction structure. Since the number of reactions does not change along with this modification, we do not need to change the kinetic law of the reaction. But due to the alteration of the reaction structure, we rename the reaction to `<oldname> variant 0`.
2. We have exactly one supporting set, as it is the case in reaction *R2*, for example (Figure 7.6, here the empty set is the only supporting set). Thus, every modifier apart from this set can be set to zero without constraining the flux of the reaction to zero. Hence, we have only to move the modifiers of the single supporting set to educt and product side. Again, we do not need to modify the kinetic law and the modified reaction will be renamed to `<oldname> variant 1`.
3. If more than one supporting set is found, a new reaction for each supporting set is derived from the original one. For each derived reaction, the respective set of modifiers of the supporting set is moved to the educt and product side, while all other stay modifiers. While the original reaction is removed, the new reactions get names like `<oldname> variant 1...<oldname> variant <number of variants>`. Since we change the number of reactions in this case, we have to adapt their kinetic laws by dividing them by the number of derived reactions. For reaction *R3* we obtain the three reactions depicted in Figure 7.7.

Abbreviation	Protein
APC	adenomatous polyposis coli
Dsh	dishevelled
ERK	extracellular signal related-kinase
GSK-3 β	glycogen synthase kinase 3 β
RKIP	Raf-1 kinase inhibitor protein
TCF	T-cell factor.

Table 7.1: Abbreviations.

7.4 Abbreviations

7.5 Detailed List Of Networks Of The BioModels Database

In the following details about the analysis of all models of the BioModels Database are given. Models in which inconsistencies have been identified are shaded in light gray. The first three rows give general details about the models. Numbers in brackets indicate the number of reactions of the original network that can increase through the processing of the kinetic laws. The number of species remains constant. The third column gives the number of reactive organization in the modified and, in brackets, of the original network. In the forth and fifth column species and reactions that can persist in a long-term simulation of the processed network are given. OT denotes the prediction by chemical organization theory and FBM the predictions by flux-based methods. In some cases FBM identifies more reactions to be present in a long-term simulation than OT. Those cases are shaded in dark-gray. The sixth and seventh column give the same numbers when inflow reactions for species with an event setting their concentration to a positive value at a certain time-point are added. In cases where the original network already contained all species, those numbers are omitted. In cases marked with (*) the heuristical approach for organization computation (11) had to be applied. Since we searched only for organizations containing the complete system, computations were aborted as soon as such an organizations was found.

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

7.5.1 Curated Branch

Table 7.2: Metabolites of the complete network

Model	Species/ Reactions	Organizations	First Step		Second Step	
			OT	FBM	OT	FBM
			(species/reactions)	(reactions)	(species/reactions)	(reactions)
BIOMD0001	12/34(34)	2(2)	12/34	34	-	-
BIOMD0002	13/34(34)	3(3)	13/34	34	-	-
BIOMD0003	3/7(7)	1(1)	3/7	7	-	-
BIOMD0004	5/7(7)	4(4)	5/7	7	-	-
BIOMD0005	9/15(15)	2(2)	9/15	15	-	-
BIOMD0006	3/5(5)	1(1)	3/5	5	-	-
BIOMD0007	22/35(35)	2(1)	22/35	35	-	-
BIOMD0008	5/13(13)	1(1)	5/13	13	-	-
BIOMD0009	22/30(30)	8(8)	22/30	30	-	-
BIOMD0010	8/10(10)	4(8)	8/10	10	-	-
BIOMD0011	22/30(30)	8(8)	22/30	30	-	-
BIOMD0012	6/12(12)	1(1)	6/12	12	-	-
BIOMD0013	27/48(48)	4(4)	27/48	48	-	-
BIOMD0014(*)	86/300(300)	53(72)	86/300	300	-	-
BIOMD0015	18/41(41)	1(1)	18/41	41	-	-
BIOMD0016	7/12(12)	1(1)	7/12	12	-	-
BIOMD0017	19/28(28)	3(3)	19/28	28	-	-
BIOMD0018	33/73(73)	1(1)	33/73	73	-	-
BIOMD0019(*)	100/256(256)	1036(1036)	100/256	256	-	-
BIOMD0021	10/26(26)	1(1)	10/26	26	-	-
BIOMD0022	13/40(40)	1(1)	13/40	40	-	-
BIOMD0023	13/38(38)	1(1)	13/38	38	-	-
BIOMD0024	3/6(6)	1(1)	3/6	6	-	-
BIOMD0025	4/10(10)	2(1)	4/10	10	-	-
BIOMD0026	11/16(16)	3(3)	11/16	16	-	-
BIOMD0027	5/8(4)	2(2)	5/8	8	-	-
BIOMD0028	16/27(27)	3(3)	16/27	27	-	-
BIOMD0029	6/11(7)	2(2)	6/11	11	-	-
BIOMD0030	18/32(32)	3(3)	18/32	32	-	-
BIOMD0031	6/10(4)	2(2)	6/10	10	-	-
BIOMD0032	37/96(96)	390(260)	37/96	96	-	-
BIOMD0033	32/56(48)	4976(8192)	32/56	56	-	-
BIOMD0034	9/22(22)	1(1)	9/22	22	-	-
BIOMD0035	10/18(18)	4(4)	10/18	18	-	-
BIOMD0036	3/7(7)	1(1)	3/7	7	-	-
BIOMD0037	12/14(14)	1(2)	3/6	10	12/14	14
BIOMD0038	17/28(28)	12(12)	17/28	28	-	-
BIOMD0039	5/7(7)	3(3)	5/7	7	-	-
BIOMD0040	5/9(9)	2(2)	5/9	9	-	-
BIOMD0041	10/17(17)	12(12)	9/17	17	9/17	17
BIOMD0042	15/34(34)	1(1)	15/34	34	-	-
BIOMD0043	5/7(7)	3(4)	5/7	7	-	-
BIOMD0044	7/8(8)	2(2)	3/4	5	6/7	7
BIOMD0045	4/8(8)	2(2)	4/8	8	-	-
BIOMD0046	16/23(23)	2(2)	16/23	23	-	-
BIOMD0047	2/3(3)	2(2)	2/3	3	-	-
BIOMD0048	23/47(47)	14(14)	23/47	47	-	-
BIOMD0049(*)	99/248(226)	3439(4513)	99/248	248	-	-
BIOMD0050	14/16(16)	1(1)	0/0	0	14/16	16
BIOMD0051	18/96(96)	1(1)	18/96	96	-	-
BIOMD0052	11/11(11)	1(1)	0/0	2	11/11	11
BIOMD0053	6/12(12)	1(1)	0/0	8	6/12	12
BIOMD0054	3/5(5)	1(1)	3/5	5	-	-
BIOMD0055	13/32(32)	1(1)	13/32	32	-	-
BIOMD0056	54/100(94)	392(16)	54/100	100	-	-
BIOMD0057	6/10(10)	2(2)	6/10	10	-	-
BIOMD0058	4/11(11)	1(1)	4/11	11	-	-
BIOMD0059	6/18(18)	2(1)	6/18	18	-	-
BIOMD0060	4/6(6)	2(2)	4/6	6	-	-
BIOMD0061	25/54(54)	8(8)	25/54	54	-	-
BIOMD0062	3/10(10)	1(1)	3/10	10	-	-
BIOMD0063	9/24(24)	1(1)	9/24	24	-	-

7.5 Detailed List Of Networks Of The BioModels Database

Model	Species/ Reactions	Organizations	First Step		Second Step	
			OT (species/reactions)	FBM (reactions)	OT (species/reactions)	FBM (reactions)
BIOMD0064	26/48(42)	16(2)	26/48	48	-	-
BIOMD0065	9/18(18)	1(1)	9/18	18	-	-
BIOMD0066	11/14(14)	13(13)	11/14	14	-	-
BIOMD0067	8/18(18)	1(1)	8/18	18	-	-
BIOMD0068	9/20(16)	1(1)	9/20	20	-	-
BIOMD0069	10/12(12)	8(8)	9/11	11	10/12	12
BIOMD0070	45/86(86)	168(168)	45/86	86	-	-
BIOMD0071	17/34(34)	3(3)	17/34	34	-	-
BIOMD0072	7/8(8)	1(2)	1/2	7	7/8	8
BIOMD0073	16/52(52)	1(1)	16/52	52	-	-
BIOMD0074	19/62(62)	1(1)	19/62	62	-	-
BIOMD0075	12/22(22)	8(8)	12/22	22	-	-
BIOMD0076	3/8(8)	1(1)	3/8	8	-	-
BIOMD0077	8/8(8)	3(3)	8/8	8	-	-
BIOMD0078	16/52(52)	1(1)	16/52	52	-	-
BIOMD0079	3/6(6)	1(1)	3/6	6	-	-
BIOMD0080	10/10(10)	3(3)	7/6	8	10/10	10
BIOMD0081	23/32(32)	18(18)	23/32	32	-	-
BIOMD0082	10/10(10)	3(3)	7/6	8	10/10	10
BIOMD0083	19/62(62)	1(1)	19/62	62	-	-
BIOMD0084	8/16(16)	16(16)	8/16	16	-	-
BIOMD0085	17/34(34)	10(10)	17/34	34	-	-
BIOMD0086	17/48(48)	12(12)	17/48	48	-	-
BIOMD0087	55/45(45)	1952(1952)	29/29	32	55/45	45
BIOMD0088	105/182(178)	936(1584)	68/122	167	105/182	182
BIOMD0089	16/41(41)	1(1)	16/41	41	-	-
BIOMD0090	26/47(47)	1(1)	26/47	47	-	-
BIOMD0091	16/25(25)	8(8)	16/25	25	-	-
BIOMD0092	4/6(6)	2(2)	4/6	6	-	-
BIOMD0093	34/48(48)	5(3)	11/16	30	31/43	43
BIOMD0094	34/47(47)	2(3)	5/5	27	24/24	40
BIOMD0095	19/46(46)	1(1)	19/46	46	-	-
BIOMD0096	19/46(46)	1(1)	19/46	46	-	-
BIOMD0097	19/46(46)	1(1)	19/46	46	-	-
BIOMD0098	2/6(6)	1(1)	2/6	6	-	-
BIOMD0099	7/14(14)	1(1)	7/14	14	-	-
BIOMD0100	5/12(12)	1(1)	5/12	12	-	-
BIOMD0101	6/13(13)	1(1)	6/13	13	-	-
BIOMD0102	13/37(37)	1(1)	13/37	37	-	-
BIOMD0103	17/61(61)	1(1)	17/61	61	-	-
BIOMD0104	6/2(2)	1(1)	1/0	0	6/2	2
BIOMD0105	39/102(102)	3(3)	13/16	49	39/102	102
BIOMD0106	25/44(44)	26(1)	14/18	40	25/44	44
BIOMD0107	14/23(23)	2(1)	14/23	23	-	-
BIOMD0108	9/20(18)	1(1)	9/20	20	-	-
BIOMD0109	61/138(138)	269(28)	48/91	128	61/138	138
BIOMD0110	15/30(30)	1(1)	15/30	30	-	-
BIOMD0111	10/20(19)	2(1)	10/20	20	-	-
BIOMD0112	10/12(12)	4(3)	4/4	11	10/12	12
BIOMD0113	4/8(8)	1(1)	4/8	8	-	-
BIOMD0114	2/5(5)	1(1)	2/5	5	-	-
BIOMD0115	2/5(5)	1(1)	2/5	5	-	-
BIOMD0116	6/10(10)	1(1)	6/10	10	-	-
BIOMD0117	2/6(6)	1(1)	2/6	6	-	-
BIOMD0119	1/1(1)	1(1)	1/1	1	-	-
BIOMD0120	5/10(10)	1(1)	3/5	9	5/10	10
BIOMD0121	6/10(10)	2(2)	6/10	10	-	-
BIOMD0122	14/38(38)	6(6)	14/38	38	-	-
BIOMD0123	14/34(34)	14(14)	14/34	34	-	-
BIOMD0124	2/2(2)	1(1)	2/2	2	-	-
BIOMD0125	5/7(7)	4(1)	5/7	7	-	-
BIOMD0126	9/22(22)	2(2)	9/22	22	-	-
BIOMD0128	3/3(3)	1(1)	3/3	3	-	-
BIOMD0137	21/32(32)	24(24)	21/32	32	-	-
BIOMD0138	1/1(1)	1(1)	1/1	1	-	-
BIOMD0139	24/64(64)	2(2)	17/39	63	24/64	64
BIOMD0140	24/64(64)	2(2)	17/39	63	24/64	64

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

Model	Species/ Reactions	Organizations	First Step		Second Step	
			OT	FBM	OT	FBM
			(species/reactions)	(reactions)	(species/reactions)	(reactions)
BIOMD0143	20/20(20)	1(1)	4/4	4	7/5	5
BIOMD0144	18/56(56)	2(1)	17/54	55	18/56	56
BIOMD0145	7/12(12)	2(1)	7/12	12	-	-
BIOMD0146	36/54(47)	29(336)	36/54	54	-	-
BIOMD0147	24/70(70)	2(2)	17/45	69	24/70	70
BIOMD0148	7/16(12)	4(1)	7/16	16	-	-
BIOMD0149	28/39(39)	150(384)	28/39	39	-	-
BIOMD0150	4/4(4)	2(2)	4/4	4	-	-
BIOMD0151	68/114(114)	80(96)	49/71	111	19/9	112
BIOMD0152	64/122(122)	143(143)	61/116	120	64/122	122
BIOMD0153	75/154(154)	147(147)	72/148	152	75/154	154
BIOMD0154	2/5(5)	1(1)	2/5	5	-	-
BIOMD0155	2/4(4)	1(1)	2/4	4	-	-
BIOMD0156	3/6(6)	1(1)	3/6	6	-	-
BIOMD0157	3/7(7)	1(1)	3/7	7	-	-
BIOMD0158	3/7(7)	1(1)	3/7	7	-	-
BIOMD0159	3/7(7)	1(1)	3/7	7	-	-
BIOMD0160	25/43(43)	32(1)	25/43	43	-	-
BIOMD0161	46/92(92)	4160(2112)	46/92	92	-	-
BIOMD0162	32/106(106)	1(1)	32/106	106	-	-
BIOMD0163	16/26(26)	4(4)	16/26	26	-	-
BIOMD0164	26/58(58)	24(12)	26/58	58	-	-
BIOMD0165	37/62(62)	624(624)	37/62	62	-	-
BIOMD0166	3/18(18)	1(1)	3/18	18	-	-
BIOMD0167	9/16(16)	4(2)	9/16	16	-	-
BIOMD0168	7/11(10)	4(1)	7/11	11	-	-
BIOMD0169	11/29(27)	1(1)	11/29	29	-	-
BIOMD0170	7/17(17)	1(1)	7/17	17	-	-
BIOMD0171	12/31(31)	1(1)	12/31	31	-	-
BIOMD0172	25/47(47)	9(9)	25/47	47	-	-
BIOMD0173	26/48(48)	16(16)	25/47	47	26/48	48
BIOMD0174	4/10(10)	2(1)	4/10	10	-	-
BIOMD0175	120/214(198)	319248(319248)	86/128	208	120/214	214
BIOMD0176	25/48(48)	9(9)	25/48	48	-	-
BIOMD0177	28/55(55)	9(9)	28/55	55	-	-
BIOMD0178	6/6(6)	1(1)	1/2	2	6/6	6
BIOMD0179	7/17(17)	1(1)	7/17	17	-	-
BIOMD0180	8/23(23)	1(1)	8/23	23	-	-
BIOMD0181	6/18(18)	1(1)	6/18	18	-	-
BIOMD0182	37/64(64)	1920(1920)	37/64	64	-	-
BIOMD0183	67/352(352)	16(16)	67/352	352	-	-
BIOMD0184	3/7(7)	1(1)	3/7	7	-	-
BIOMD0185	8/20(20)	1(1)	8/20	20	-	-

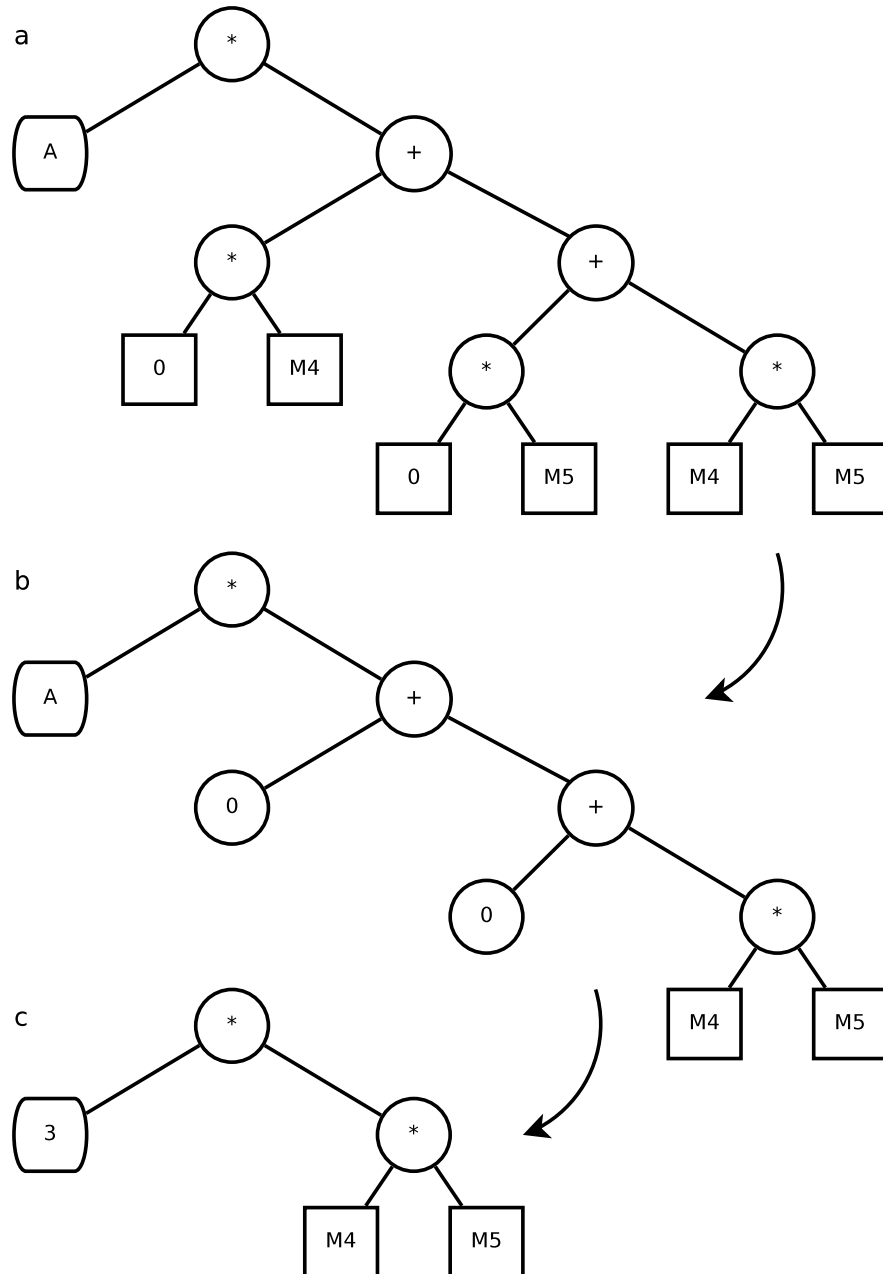


Figure 7.5: In (a), we applied a zero concentration to $\{M3\}$, solved the multiplications by zero (b) and obtained the rate law $v_{R3} = 3 \cdot [M4] \cdot [M5]$ after the application of A 's value 3 (c).

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

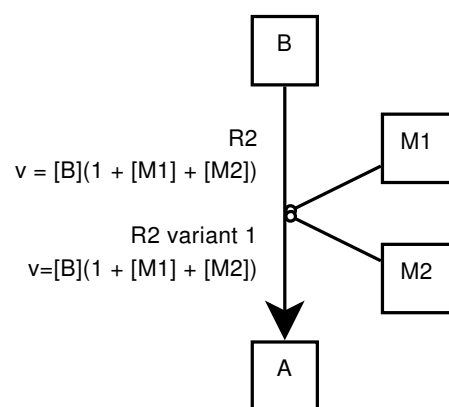


Figure 7.6: Since the supporting set is the empty set, no modifier has to be moved to the educt or product side, hence also the kinetic laws are left unchanged.

7.5 Detailed List Of Networks Of The BioModels Database

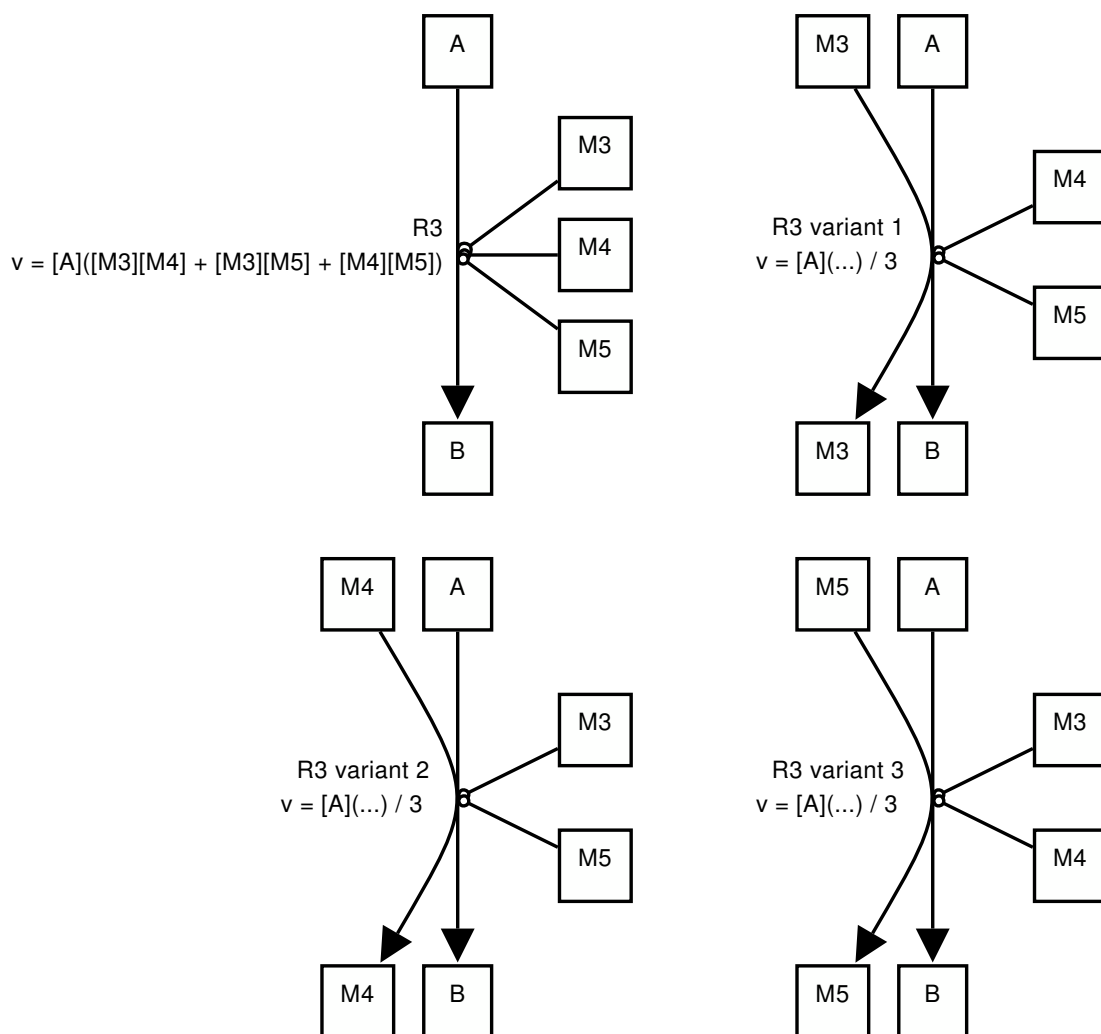


Figure 7.7: We obtain three supporting modifier sets, leading to three reaction variants. Note the division by 3 in the kinetic laws.

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

7.5.2 Non-Curated Branch

The non-curated branch of the BioModels Database contains 74 models. Of these, we could validate 55 with the same methods as for the curated branch. For 18 networks we were not able to compute the entire set of organizations. This was caused by too large a set of organizations in 9 cases ($> 10^6$ organizations) in which we had to abort computation due to constraints in memory. However, in these cases we already found an organization encompassing the whole species set. The remaining 9 networks were too large for a detailed analysis since they contained more than 500 reactions. Centler (11) analyzed such a network by applying a heuristic whose results need to be carefully checked. This is due to the nature of the heuristic algorithm which only approximates the set of organizations. Hence, some organizations might not be found if the heuristic is aborted too early. Since these models are contained in the non-curated branch of the BioModels Database, which is not the central focus of this work, and error-checking is time-consuming we applied only flux-based methods in these cases. Finally, we could not open *MODEL8262229752* with the JigCell SBML parser. The parser indicated that the model contains a syntactic error. The cases where we could compute chemical organizations are listed next. Computations were performed as described in the previous section.

Table 7.3: Metabolites of the complete network

Model	Species/ Reactions	Organizations	First Step		Second Step	
			OT (species/reactions)	FBM (reactions)	OT (species/reactions)	FBM (reactions)
MODEL0212154960	5/10(10)	1(1)	0/0	6	3/5	7
MODEL0995500644	12/13(13)	1(1)	11/12	12	12/13	13
MODEL1502077979	7/13(13)	1(1)	7/13	13	-	-
MODEL2463576061	330/222(222)	48(48)	33/14	22	33/14	22
MODEL2463683119	680/470(470)	256(256)	71/26	40	71/26	40
MODEL2504064544	19/52(51)	1(1)	19/52	52	-	-
MODEL4665428627	61/146(146)	6(6)	9/12	96	61/146	146
MODEL4734733125	14/24(24)	4(4)	14/24	24	-	-
MODEL4779732381	14/17(17)	4(4)	11/14	16	14/17	17
MODEL4780441670	8/11(11)	2(2)	5/8	10	8/11	11
MODEL4780784080	14/24(24)	2(2)	10/21	23	14/24	24
MODEL4816599063	12/18(18)	4(4)	12/18	18	-	-
MODEL4821294342	12/26(26)	2(2)	12/26	26	-	-
MODEL4968912141	8/10(10)	3(3)	8/10	10	-	-
MODEL4969417017	18/28(28)	11(11)	15/22	26	18/28	28
MODEL5073396359	22/70(70)	2(2)	7/16	59	9/17	60
MODEL5662324959	628/2212(2212)	1(1)	628/2212	2212	-	-
MODEL5662377562	628/2212(2212)	1(1)	628/2212	2212	-	-
MODEL5662398146	628/2212(2212)	1(1)	628/2212	2212	-	-
MODEL5662425708	628/2212(2212)	1(1)	628/2212	2212	-	-
MODEL5974712823	10/6(6)	2(2)	8/5	5	8/6	6
MODEL6623617994	22/36(36)	22(22)	22/36	36	-	-
MODEL6623628741	10/8(8)	2(2)	10/8	8	-	-
MODEL6624091635	34/80(80)	14(14)	34/80	80	-	-
MODEL6624199343	5/10(10)	2(2)	5/10	10	-	-
MODEL6762427183	0/0(0)	1(1)	0/0	0	-	-
MODEL8568434338	225/219(219)	1(1)	28/0	0	28/0	0
MODEL8583955822	12/30(30)	1(1)	12/30	30	-	-
MODEL8584137422	12/30(30)	1(1)	12/30	30	-	-

7.5 Detailed List Of Networks Of The BioModels Database

Model	Species/ Reactions	Organizations	First Step		Second Step	
			OT (species/reactions)	FBM (reactions)	OT (species/reactions)	FBM (reactions)
MODEL8584292730	13/38(38)	1(1)	13/38	38	-	-
MODEL8584468482	13/36(36)	1(1)	13/36	36	-	-
MODEL8938094216	15/18(18)	19(24)	15/18	18	-	-
MODEL9070467164	94/179(179)	4656(4656)	94/179	179	-	-
MODEL9071122126	64/116(116)	280(280)	64/116	116	-	-
MODEL9071773985	73/147(147)	380(380)	73/147	147	-	-
MODEL9077438479	29/48(48)	78(90)	29/48	48	-	-
MODEL9079179924	81/146(146)	4512(4512)	81/146	146	-	-
MODEL9079740062	29/48(48)	78(90)	29/48	48	-	-
MODEL9080388197	15/26(26)	2(2)	15/26	26	-	-
MODEL9080747936	50/90(90)	288(288)	50/90	90	-	-
MODEL9081220742	188/350(350)	$> 10^6 (> 10^6)$	188/350	350	-	-
MODEL9085850385	59/104(104)	640(640)	59/104	104	-	-
MODEL9086207764	284/580(580)	$> 10^6 (> 10^6)$	284/580	580	-	-
MODEL9086518048	286/594(594)	$> 10^6 (> 10^6)$	286/594	594	-	-
MODEL9086628127	16/32(32)	3(3)	16/32	32	-	-
MODEL9086926384	85/156(156)	1112(1112)	85/156	156	-	-
MODEL9086953089	114/206(206)	43245(46416)	114/206	206	-	-
MODEL9087255381	289/602(602)	$> 10^6 (> 10^6)$	289/602	602	-	-
MODEL9087474843	290/602(602)	$> 10^6 (> 10^6)$	290/602	602	-	-
MODEL9087766308	5/4(4)	2(2)	5/4	4	-	-
MODEL9087988095	5/4(4)	2(2)	5/4	4	-	-
MODEL9088169066	5/4(4)	2(2)	5/4	4	-	-
MODEL9088294310	5/4(4)	2(2)	5/4	4	-	-
MODEL9089491423	196/364(364)	$> 10^6 (> 10^6)$	196/364	364	-	-
MODEL9089538076	200/374(374)	$> 10^6 (> 10^6)$	200/374	374	-	-
MODEL9089914876	192/358(358)	$> 10^6 (> 10^6)$	192/358	358	-	-
MODEL9147091146	77/142(142)	752(752)	77/142	142	-	-
MODEL9147232940	64/112(112)	996(996)	64/112	112	-	-
MODEL9147975215	37/49(49)	72(72)	35/46	48	37/49	49
MODEL9200487367	5/9(9)	2(2)	5/9	9	-	-
MODEL9852292468	73/66(66)	1(1)	0/0	0	61/52	56

For the remaining 9 cases we adapted a different approach by just checking whether the entire species set is an organization. If we did not find this set to be an organization, we added, similar to the original approach, an inflow for every species whose concentration is set to a non-zero value at a certain time-point. In contrast, we were able to test whether each reaction could be present in a steady-state or growth state flux using the emulation method for flux-based methods described in Section 7.1.

Model	Species/ Reactions	Organizations	First Step		Second Step	
			OT (species/reactions)	FBM (reactions)	OT (species/reactions)	FBM (reactions)
MODEL0403888565	377/805(805)	n.a.	n.a.	445	377/805	805
MODEL0403928902	377/805(805)	n.a.	n.a.	445	377/805	805
MODEL0403954746	377/805(805)	n.a.	n.a.	445	377/805	805
MODEL0403988150	377/806(806)	n.a.	n.a.	446	377/805	805
MODEL0404023805	377/806(806)	n.a.	n.a.	446	377/805	805
MODEL2021729243	2715/4370(4370)	n.a.	n.a.	3733	n.a.	3733
MODEL2021747594	2715/4370(4370)	n.a.	n.a.	3592	n.a.	3592
MODEL3023609334	1972/3842(3842)	n.a.	n.a.	3752	n.a.	3752
MODEL3023641273	1972/3842(3842)	n.a.	n.a.	3752	n.a.	3752
MODEL4132046015	408/534(534)	n.a.	n.a.	32	n.a.	32

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

7.5.3 Discussion

Next, we will briefly discuss the models of the non-curated branch of the BioModels Database in which we detected inconsistencies. Overall we found 11 models containing inconsistencies. However, a first examination reveals that in 6 of these models, *MODEL2021729243*, *MODEL2021747594*, *MODEL2463576061*, *MODEL2463683119*, *MODEL4132046015* and *MODEL8568434338*, neither an input nor an initial concentration for any species is given. Thus, it can be assumed that these models have been constructed for the purpose of a structural and not a dynamic analysis.

Another two models, *MODEL3023609334* and *MODEL3023641273*, represent a genome-scale reconstruction of the metabolism of *E. coli*. Since the metabolism of *E. coli* is not yet entirely understood, it contains species which are only consumed and thus also the reactions using them as educts cannot have a positive flux. Most of the reactions and species that cannot be present in the limit behavior are due to this missing knowledge. Additionally, some species and reactions are not present because they belong to uptake and utilization pathways for metabolites that are not contained in the growth media.

The remaining models are *MODEL0212154960*, modeling the vectorial transport of bromosulphophthalein over epithelial cells (3), *MODEL9852292468*, modeling lipid-mediated thrombin generation (8), and *MODEL5073396359*, modeling apoptosome-dependent caspase activation (55). Comparing *MODEL0212154960* to the model presented in Bartholomé (3), we found that a reaction was missing in the SBML model, while it was described in the supplementary material of the publication. Adding this reaction, we found all species present in an organization. During simulation, the two species absent from any organization had indeed a zero concentration. In *MODEL9852292468* we found reactions that had a negative flux during simulation while they were set to irreversible in the model. Relaxing the irreversibility constraint in these cases we found all species present in an organization. Finally, analyzing *MODEL5073396359* and comparing to the supplementary material of Rehm (55) we found that several species were not supplied as input in the SBML model, while the description of the model in the publication contained such an inflow. In this case, even the dynamic behavior of the SBML model did not match the behavior of the model described in the original publication. Adding an inflow or an initial concentration resolved the inconsistencies. In all three inconsistent models we found that chemical organization theory predicted the reactions that can have a non-zero flux during simulation more accurately than flux-based methods. The reasons for these differences follow the scheme outlined in the main article. Thus, flux-based methods find a steady state flux

7.5 Detailed List Of Networks Of The BioModels Database

through a set of reactions encompassing species that can not be produced at positive rate. However, due to a drain of one or several of these species by interconversion to other species or decay, they cannot persist in a long-term simulation. Since OT takes this drain into account, the species are found absent from any organization.

7. SUPPLEMENT: USING CHEMICAL ORGANIZATION THEORY FOR MODEL-CHECKING

8

SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

Stephan Richter, Ingo Fetzer, Florian Centler, Peter Dittrich, Martin Thullner

Here we list the tables of errors as of may 2013.

8.1 Same Substances With Different Formulas

The following table lists KEGG compound, glycan, and drug entries, which are supposed to be “same”, but show different chemical formulas.

Table 8.1: Dissent Formulas in synonym Compound Entries

KEGG Entry	formulas	comment
C00369 / C00721 / C03018 / D00084 / D06507 / G10545	$C_{30}H_{51}O_{26}$ / $C_{60}H_{100}O_{50}$	n replaced by 5
C00670 / D07349	$C_8H_{20}NO_6P$ / $C_8H_{21}NO_6P$	ionized/recombined form
C00718 / C01935 / D02329 / G10495	$C_{30}H_{50}O_{25}$ / $C_{60}H_{100}O_{50}$	n replaced by 5, basic unit doubled
C00734 / C06023 / G10536	$C_{42}H_{79}N_7O_{25}$ / $C_{42}H_{79}N_7O_{29}$	n replaced by 5

continued on next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page		
KEGG Entry	formulas	comment
C01768 / D02324 / G07287 / G07471 / G08476 / G10593	$C_{24}H_{34}O_{25} / C_{30}H_{40}O_{30}$	n replaced by 5, formula resolved via KCF and KEGG Codes table
C02399 / G00154	$C_8H_{15}NO_7 / C_9H_{14}N_2O_7R_2$	formula resolved via KCF and KEGG Codes table
C04276 / C04772 / G10509	$C_{72}H_{121}O_{65} / C_{78}H_{131}O_{66}$	n replaced by 5, connected via glycan
C04750 / C04776 / G00024	$C_{14}H_{24}NO_{11}R / C_{18}H_{29}N_3O_{13}R_2$	connected via glycan
C04825 / G00156	$C_{20}H_{35}NO_{17} / C_{21}H_{34}N_2O_{17}R_2$	formula resolved via KCF and KEGG Codes table
C04903 / G00157	$C_{26}H_{43}NO_{23} / C_{27}H_{42}N_2O_{23}R_2$	formula resolved via KCF and KEGG Codes table
C07283 / G00830	$C_{18}H_{32}O_{16} / C_{42}H_{72}O_{40}$	only equal for n=1
C15656 / G00162	$C_{34}H_{56}N_2O_{28} / C_{35}H_{55}N_3O_{28}R_2$	formula resolved via KCF and KEGG Codes table

The following tables list the sets of indistinctive, unbalanced and “transmutational” reactions.

8.2 Indistinctive Reactions

Table 8.2: Indistinctive reactions

Reaction Code	Substance without chemical formula
R02592, R04774	9005 (Activated methyl group)
R05512	9005 (Activated methyl group), 8533 (Amino group donor)
R09740	844 (2-Hydroxy fatty acid)
R09741	843 (3-Hydroxy fatty acid)
R09742	15981 (Cyclic alcohol), 4985 (Quinone), 4984 (Hydroquinone), 842 (Cyclic ketone)
R01237, R01501	8414 (Sugar)
R00804, R03076	18010 (Sugar phosphate), 8414 (Sugar)
R05777	8368 (Diphospho-myo-inositol polyphosphate),

continued on next page

8.2 Indistinctive Reactions

continued from previous page	
Reaction Code	Substance without chemical formula
	8367 (myo-Inositol polyphosphate)
R06137	14133 (Cyclic amidines), 8034 (Cyclic amide)
R07132	5102 (D-Galactonolactone)
R07192	13773 (Ketosteroid), 5080 (Steroid ester)
R07193	13773 (Ketosteroid), 5079 (Steroid lactone)
R07343	5001 (myo-Inositol phosphate)
R03754	15724 (Acyl-sn-glycerol 3-phosphate), 4996 (Acylglycerol)
R07347	18908 (Hydrogen-acceptor), 18905 (Hydrogen-donor), 4994 (Pyranose), 4993 (2-Dehydropyranose)
R07348	18908 (Hydrogen-acceptor), 18905 (Hydrogen-donor), 4992 (Pyranoside), 4991 (3-Dehydropyranoside)
R07349, R07350	4990 (n-Alkanal), 4989 (Alk-2-enal)
R07356, R07357	4986 (Glyceollin)
R02364	14707 (Semiquinone), 4985 (Quinone)
R00849, R01868, R06247, R02365, R07358, R07359, R07361, R07511, R09322, R09493, R09494, R09497, R09518, R09656, R09658	4985 (Quinone), 4984 (Hydroquinone)
R04007	4780 (p-Hydroxyphenyl lignin)
R02596	4779 (Guaiacyl lignin)
R03919	4778 (Syringyl lignin)
R07443	4777 (5-Hydroxy-guaiacyl lignin)
R07459	4774 (Thiamine biosynthesis intermediate 1), 4771 (Thiamine biosynthesis intermediate 4)
R10247	4774 (Thiamine biosynthesis intermediate 1), 4770 (Thiamine biosynthesis intermediate 5)
R07461	4771 (Thiamine biosynthesis intermediate 4), 4770 (Thiamine biosynthesis intermediate 5)
R07464, R07465	4769 (Thiamine biosynthesis intermediate 6)
R07462	4770 (Thiamine biosynthesis intermediate 5), 4769 (Thiamine biosynthesis intermediate 6)
R07612	4594 (Oxidized reactive black 5)
R07646, R08578	4459 (tRNA(Pyl))
R08146	4109 (Farnesal)

continued on next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page	
Reaction Code	Substance without chemical formula
R08145	17551 (Farnesol), 4109 (Farnesal)
R08218	3976 (tRNA(Sec))
R08360, R08361, R08363	3950 (Tryparedoxin), 3949 (Tryparedoxin disulfide)
R08386	3931 (N-Acetoxyarylamine), 3930 (N-Hydroxyarylamine)
R02847	3787 (D-Aldonolactone)
R07767, R07768, R09395	3614 (Sulfur donor)
R01078	14666 (Electron), 3614 (Sulfur donor)
R03541	3125 (Prenyl diphosphate), 3124 (Prenol)
R06238	29732 ((GlcNAc)2 (Man)3 (PP-Dol)1)
R05976	29731 ((Glc)3 (GlcNAc)2 (Man)9 (PP-Dol)1)
R05988	29723 ((GlcNAc)4 (LFuc)1 (Man)3 (Asn)1)
R05989	29723 ((GlcNAc)4 (LFuc)1 (Man)3 (Asn)1), 29722 ((Gal)2 (GlcNAc)4 (LFuc)1 (Man)3 (Asn)1)
R05990	29722 ((Gal)2 (GlcNAc)4 (LFuc)1 (Man)3 (Asn)1), 29721 (DS 3)
R05907, R05908, R05909	29716 (Tn antigen)
R05914	29715 ((Gal)1 (GalNAc)1 (Neu5Ac)2 (Ser/Thr)1)
R05910	29716 (Tn antigen), 29713 ((GalNAc)1 (GlcNAc)1 (Ser/Thr)1)
R07628	29713 ((GalNAc)1 (GlcNAc)1 (Ser/Thr)1), 29712 ((Gal)1 (GalNAc)1 (GlcNAc)1 (Ser/Thr)1)
R05911	29716 (Tn antigen), 29709 (Sialyl-Tn antigen)
R06164	29707 ((Gal)3 (Glc)1 (GlcNAc)1 (LFuc)2 (Cer)1)
R06167	29706 (Type IA glycolipid)
R06162	29706 (Type IA glycolipid), 29705 ((Gal)2 (GalNAc)1 (Glc)1 (GlcNAc)1 (LFuc)2 (Cer)1)
R06163	29704 (Leb glycolipid)
R06155	29703 (Lea glycolipid)
R06165	29701 (Fuc-3'-isoLM1)
R06027	29699 (Type II B antigen)
R06024, R06187	29697 (Type IIIH glycolipid)
R06029	29699 (Type II B antigen), 29697 (Type IIIH glycolipid)
R06095	29697 (Type IIIH glycolipid), 29696 (Ley glycolipid)
R06198	29695 ((Gal)3 (GalNAc)1 (Glc)1 (GlcNAc)1 (LFuc)1 (Cer)1)
R06031	29695 ((Gal)3 (GalNAc)1 (Glc)1 (GlcNAc)1 (LFuc)1 (Cer)1), 29694 (Type IIIH glycolipid)
R06197	29694 (Type IIIH glycolipid), 29693 (Type IIIA glycolipid)

continued on next page

8.2 Indistinctive Reactions

continued from previous page	
Reaction Code	Substance without chemical formula
R06025	29692 (Lacto-N-fucopentaosyl III ceramide)
R06075	29690 (IV3-a-NeuAc,III3-a-Fuc-nLc4Cer)
R06035	29684 (VI2Fuc-nLc6)
R06193	29684 (VI2Fuc-nLc6), 29683 ((Gal)3 (GalNAc)1 (Glc)1 (GlcNAc)2 (LFuc)1 (Cer)1)
R06192	29683 ((Gal)3 (GalNAc)1 (Glc)1 (GlcNAc)2 (LFuc)1 (Cer)1), 29682 ((Gal)4 (GalNAc)1 (Glc)1 (GlcNAc)2 (LFuc)1 (Cer)1)
R06041	29682 ((Gal)4 (GalNAc)1 (Glc)1 (GlcNAc)2 (LFuc)1 (Cer)1), 29681 ((Gal)4 (GalNAc)1 (Glc)1 (GlcNAc)2 (LFuc)2 (Cer)1)
R06191	29681 ((Gal)4 (GalNAc)1 (Glc)1 (GlcNAc)2 (LFuc)2 (Cer)1), 29680 (Type IIIAb)
R06230	29679 (III3Fuc-nLc6Cer)
R06076	29684 (VI2Fuc-nLc6), 29674 ((Gal)3 (Glc)1 (GlcNAc)2 (LFuc)2 (Cer)1)
R06227	29674 ((Gal)3 (Glc)1 (GlcNAc)2 (LFuc)2 (Cer)1), 29673 ((Gal)3 (Glc)1 (GlcNAc)2 (LFuc)3 (Cer)1)
R06190	29684 (VI2Fuc-nLc6), 29672 ((Gal)4 (Glc)1 (GlcNAc)2 (LFuc)1 (Cer)1)
R06039	29671 ((Gal)4 (Glc)1 (GlcNAc)3 (LFuc)1 (Cer)1)
R06222	29671 ((Gal)4 (Glc)1 (GlcNAc)3 (LFuc)1 (Cer)1), 29670 ((Gal)4 (Glc)1 (GlcNAc)3 (LFuc)2 (Cer)1)
R06224	29670 ((Gal)4 (Glc)1 (GlcNAc)3 (LFuc)2 (Cer)1), 29669 ((Gal)4 (Glc)1 (GlcNAc)3 (LFuc)3 (Cer)1)
R06038	29666 (V3Fuc-nLc6Cer)
R06221	29666 (V3Fuc-nLc6Cer), 29665 (V3Fuc,III3Fuc-nLc6Cer)
R06232	29666 (V3Fuc-nLc6Cer), 29664 ((Gal)3 (Glc)1 (GlcNAc)2 (LFuc)1 (Neu5Ac)1 (Cer)1)
R05968	29660 (Globo-H)
R05904	29655 ((Gal)4 (GalNAc)1 (Glc)1 (GlcNAc)1 (LFuc)1 (Cer)1)
R05923	29639 ((GlcN)1 (Ino(acyl)-P)1 (Man)4 (EtN)1 (P)1), 29638 ((GlcN)1 (Ino(acyl)-P)1 (Man)4 (EtN)2 (P)2)
R05916, R02654	29636 ((GlcNAc)1 (Ino-P)1)
R07398, R06623	29635 ((GlcN)1 (Ino-P)1)
R05917, R03482	29636 ((GlcNAc)1 (Ino-P)1), 29635 ((GlcN)1 (Ino-P)1)
R05918	29635 ((GlcN)1 (Ino-P)1), 29634 ((GlcN)1 (Ino(acyl)-P)1)
R05919	29634 ((GlcN)1 (Ino(acyl)-P)1), 29633 ((GlcN)1 (Ino(acyl)-P)1 (Man)1)
R05920	29633 ((GlcN)1 (Ino(acyl)-P)1 (Man)1),

continued on next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page	
Reaction Code	Substance without chemical formula
R05921	29632 ((GlcN)1 (Ino(acyl)-P)1 (Man)1 (EtN)1 (P)1) 29632 ((GlcN)1 (Ino(acyl)-P)1 (Man)1 (EtN)1 (P)1), 29631 ((GlcN)1 (Ino(acyl)-P)1 (Man)2 (EtN)1 (P)1)
R07129	29639 ((GlcN)1 (Ino(acyl)-P)1 (Man)4 (EtN)1 (P)1), 29630 ((GlcN)1 (Ino(acyl)-P)1 (Man)3 (EtN)1 (P)1)
R05922	29631 ((GlcN)1 (Ino(acyl)-P)1 (Man)2 (EtN)1 (P)1), 29630 ((GlcN)1 (Ino(acyl)-P)1 (Man)3 (EtN)1 (P)1)
R05924	29638 ((GlcN)1 (Ino(acyl)-P)1 (Man)4 (EtN)2 (P)2), 29629 ((GlcN)1 (Ino(acyl)-P)1 (Man)4 (EtN)3 (P)3)
R05944	29619 (Fucosyl-GM1)
R06015	29613 ((GlcNAc)4 (LFuc)1 (Man)3 (Asn)1)
R08849	2834 (Phaeomelanin)
R04886, R06612	2833 (Eumelanin)
R06283	29488 (3'-LM1-NeuGc), 27292 (nLc4)
R08964	2713 (1D-myo-Inositol bisdiphosphate tetrakisphosphate)
R06274	26285 ((Gal)2 (GalNAc)3 (Neu5Ac)2 (Neu5Gc)1 (Ser/Thr)1), 25984 ((Gal)2 (GalNAc)3 (Neu5Ac)1 (Neu5Gc)1 (Ser/Thr)1)
R06085	25903 (Monofucosyllactoisooctaosylceramide)
R06086	29676 ((Gal)4 (Glc)1 (GlcNAc)3 (LFuc)2 (Cer)1), 25903 (Monofucosyllactoisooctaosylceramide)
R06273	25984 ((Gal)2 (GalNAc)3 (Neu5Ac)1 (Neu5Gc)1 (Ser/Thr)1), 25635 ((Gal)2 (GalNAc)3 (Neu5Gc)1 (Ser/Thr)1)
R09127, R09128	2542 (Cytochrome cL), 2541 (Reduced cytochrome cL)
R06089	20783 (Monofucosyllactoisooctaosylceramide)
R06090	29676 ((Gal)4 (Glc)1 (GlcNAc)3 (LFuc)2 (Cer)1), 20783 (Monofucosyllactoisooctaosylceramide)
R06128	20782 ((GlcNAc)2 (Man)4 (PP-Dol)1)
R06127	29732 ((GlcNAc)2 (Man)3 (PP-Dol)1), 20782 ((GlcNAc)2 (Man)4 (PP-Dol)1)
R06172	20781 ((GlcNAc)1 (MurNAc)1 (D-Ala-D-Ala-Lys-D-Glu-Ala)1 (PP-Und)1), 20780 ((MurNAc)1 (D-Ala-D-Ala-Lys-D-Glu-Ala)1 (PP-Und)1)
R06258	20779 ((GlcNAc)2 (Man)6 (PP-Dol)1)
R06259	20779 ((GlcNAc)2 (Man)6 (PP-Dol)1), 20778 ((GlcNAc)2 (Man)7 (PP-Dol)1)
R06261	20777 ((GlcNAc)2 (Man)8 (PP-Dol)1)
R06260	20778 ((GlcNAc)2 (Man)7 (PP-Dol)1), 20777 ((GlcNAc)2 (Man)8 (PP-Dol)1)
R06262	20776 ((Glc)1 (GlcNAc)2 (Man)9 (PP-Dol)1)

continued on next page

8.2 Indistinctive Reactions

continued from previous page	
Reaction Code	Substance without chemical formula
R06264	29731 ((Glc)3 (GlcNAc)2 (Man)9 (PP-Dol)1), 20775 ((Glc)2 (GlcNAc)2 (Man)9 (PP-Dol)1)
R06263	20776 ((Glc)1 (GlcNAc)2 (Man)9 (PP-Dol)1), 20775 ((Glc)2 (GlcNAc)2 (Man)9 (PP-Dol)1)
R09319	20688 ((GlcNAc)3 (LFuc)1 (Man)3 (Asn)1)
R09301	20708 ((Man)2 (Ser/Thr)1), 20648 ((Man)3 (Ser/Thr)1)
R09302	20648 ((Man)3 (Ser/Thr)1), 20647 ((Man)4 (Ser/Thr)1)
R09303	20647 ((Man)4 (Ser/Thr)1), 20565 ((Man)5 (Ser/Thr)1)
R07617	20539 ((Gal)2 (GalNAc)1 (GlcNAc)1 (Ser/Thr)1)
R09304	19345 ((GlcNAc)1 (Ser/Thr)1)
R07614	18982 ((Gal)1 (GalNAc)2 (GlcNAc)1 (Ser/Thr)1)
R07620, R04072	18978 ((Man)1 (Ser/Thr)1)
R09300	20708 ((Man)2 (Ser/Thr)1), 18978 ((Man)1 (Ser/Thr)1)
R07619	18979 ((GlcNAc)1 (Man)1 (Ser/Thr)1), 18978 ((Man)1 (Ser/Thr)1)
R07621	18979 ((GlcNAc)1 (Man)1 (Ser/Thr)1), 18977 ((GlcNAc)2 (Man)1 (Ser/Thr)1)
R07811	18971 ((GlcA)2 (GlcN)1 (GlcNAc)1 (LIdoA)1 (S)4)
R07813	18970 ((GlcA)2 (GlcN)1 (GlcNAc)1 (LIdoA)1 (S)3)
R07812	18971 ((GlcA)2 (GlcN)1 (GlcNAc)1 (LIdoA)1 (S)4), 18970 ((GlcA)2 (GlcN)1 (GlcNAc)1 (LIdoA)1 (S)3)
R07820	18964 ((GalNAc)2 (GlcA)1 (LIdoA)1 (S)3)
R07822	18963 ((GalNAc)2 (GlcA)1 (LIdoA)1 (S)2)
R07821	18964 ((GalNAc)2 (GlcA)1 (LIdoA)1 (S)3), 18963 ((GalNAc)2 (GlcA)1 (LIdoA)1 (S)2)
R08107	29630 ((GlcN)1 (Ino(acyl)-P)1 (Man)3 (EtN)1 (P)1), 18961 ((GlcN)1 (Ino(acyl)-P)1 (Man)3 (EtN)2 (P)2)
R09295	18953 ((LFuc)1 (Ser/Thr)1)
R09316	20599 ((Glc)1 (LFuc)1 (Ser/Thr)1), 18953 ((LFuc)1 (Ser/Thr)1)
R09296	18953 ((LFuc)1 (Ser/Thr)1), 18952 ((GlcNAc)1 (LFuc)1 (Ser/Thr)1)
R09297	24681 ((Gal)1 (GlcNAc)1 (LFuc)1 (Neu5Ac)1 (Ser/Thr)1), 18951 ((Gal)1 (GlcNAc)1 (LFuc)1 (Ser/Thr)1)
R09298	20598 ((Gal)1 (GlcNAc)1 (LFuc)1 (Neu5Ac)1 (Ser/Thr)1), 18951 ((Gal)1 (GlcNAc)1 (LFuc)1 (Ser/Thr)1)
R09299	18952 ((GlcNAc)1 (LFuc)1 (Ser/Thr)1), 18951 ((Gal)1 (GlcNAc)1 (LFuc)1 (Ser/Thr)1)
R09318	18949 ((GlcNAc)4 (LFuc)1 (Man)3 (Xyl)1 (Asn)1)

continued on next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page	
Reaction Code	Substance without chemical formula
R09320	20688 ((GlcNAc)3 (LFuc)1 (Man)3 (Asn)1), 18948 ((Gal)1 (GlcNAc)3 (LFuc)1 (Man)3 (Asn)1)
R09323	18948 ((Gal)1 (GlcNAc)3 (LFuc)1 (Man)3 (Asn)1), 18947 ((Gal)1 (GlcNAc)2 (LFuc)1 (Man)3 (Asn)1)
R09324	18947 ((Gal)1 (GlcNAc)2 (LFuc)1 (Man)3 (Asn)1), 18946 ((Gal)1 (GlcNAc)2 (LFuc)2 (Man)3 (Asn)1)
R00047, R00073, R00074, R00280, R00281, R00282, R00283, R00284, R00295, R00296, R00297, R00298, R00305, R00311, R00326, R00361, R00374, R00412, R00476, R00544, R00609, R00638, R00639, R00645, R00798, R00808, R00860, R00861, R00873, R00976, R01025, R01045, R01253, R01282, R01303, R01306, R01342, R01374, R01413, R01508, R01599, R01696, R01697, R01742, R01833, R01834, R01854, R01915, R02166, R02206, R02211, R02212, R02213, R02214, R02215, R02234, R02264, R02374, R02612, R02642, R02643, R02661, R02838, R02860,	18908 (Hydrogen-acceptor), 18905 (Hydrogen-donor)
continued on next page	

8.2 Indistinctive Reactions

continued from previous page	
Reaction Code	Substance without chemical formula
R02873, R02987, R03015, R03086, R03156, R03172, R03173, R03185, R03195, R03206, R03212, R03257, R03326, R03370, R03441, R03510, R03532, R03533, R03575, R03597, R03599, R03687, R03714, R03724, R03734, R03748, R03783, R03784, R03793, R03814, R03833, R03849, R03927, R04080, R04160, R04178, R04327, R04392, R04437, R04571, R04622, R04667, R04693, R04760, R04786, R04787, R04798, R04799, R04800, R04803, R04827, R04852, R04947, R04971, R04973, R04979, R05040, R05059, R05060, R05084, R05151, R05152, R05183, R05255, R05260, R05346, R05579, R05583, R05619, R05704, R05708, R05740, R05742, R05745, R05752, R05753,	
continued on next page	

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page	
Reaction Code	Substance without chemical formula
R06219, R01680, R06246, R01443, R06268, R06269, R06270, R06306, R06370, R06372, R06373, R06401, R06402, R06403, R06519, R07063, R07153, R07154, R07155, R07163, R07166, R07167, R07174, R07182, R07218, R07223, R07229, R07230, R07374, R07431, R07467, R07470, R07514, R07518, R07520, R07531, R07654, R07850, R07861, R07933, R07946, R08089, R08161, R08173, R08517, R08518, R08701, R08735, R08740, R08763, R09156, R09293, R09481, R09496, R09519, R09551, R09604, R09605, R09659, R09671, R09691, R09692, R09693, R09703, R09716, R09727, R09728, R09884, R10083, R10085, R10193, R10246	
R00019, R00021, R00790, R00791,	18802 (Reduced ferredoxin), 18801 (Oxidized ferredoxin)
continued on next page	

8.2 Indistinctive Reactions

continued from previous page	
Reaction Code	Substance without chemical formula
R00859, R01195, R01196, R01197, R01199, R01217, R02218, R02451, R02550, R02675, R02843, R03164, R03329, R03569, R03600, R03678, R03851, R04850, R05185, R05316, R05496, R05739, R05817, R05818, R05819, R05875, R06282, R07157, R07159, R07160, R07161, R07409, R07525, R07526, R07537, R08566, R08567, R08571, R08689, R09053, R09060, R09071, R09486, R09491, R09502, R09508, R09587, R10086, R10158, R10162	
R00389, R00394, R07325 R07158	18765 (Acid) 18802 (Reduced ferredoxin), 18801 (Oxidized ferredoxin), 18765 (Acid)
R02128, R02905, R03137, R03311, R03536, R03677, R04036, R04063, R04078, R07128, R09503, R09571 R03907	18733 (Photon) 18908 (Hydrogen-acceptor), 18905 (Hydrogen-donor), 18733 (Photon)
R00329, R01532,	18719 (Nucleotide)
continued on next page	

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page	
Reaction Code	Substance without chemical formula
R01572, R07341	
R01347, R02000, R02002, R02281, R02879, R05489, R05490, R05491, R05743	18591 (Reduced rubredoxin), 18493 (Oxidized rubredoxin)
R00162, R00164	18361 (Phosphoprotein)
R01454, R02723, R02724, R02725, R03262, R03263, R03933, R04676, R04854, R04855, R08949, R10159	18275 (Reduced adrenal ferredoxin), 18270 (Oxidized adrenal ferredoxin)
R02806, R03852, R04168	18237 (Nucleobase)
R02918	18155 (tRNA(Tyr))
R00106, R03071, R09473	18020 (Ferricytochrome), 18019 (Ferrocyclochrome)
R03149, R03570	17956 (Monosaccharide 1-phosphate)
R03146	17952 (Ferricytochrome b1), 17949 (Ferrocyclochrome b1)
R00100, R01115, R01803, R02222, R03147, R08539	17951 (Ferricytochrome b5), 17948 (Ferrocyclochrome b5)
R00082, R00108, R00784, R09500	17950 (Ferricytochrome c2), 17947 (Ferrocyclochrome c2)
R02884, R02885	18181 (Caldesmon), 17927 (Caldeumon phosphate)
R02726	18551 (Steroid), 18275 (Reduced adrenal ferredoxin), 18270 (Oxidized adrenal ferredoxin), 17898 (11beta-Hydroxysteroid)
R00198, R03215	17887 (Ferricytochrome c-553), 17886 (Ferrocyclochrome c-553)
R07824	17884 (N-Acetylgalactosamine)
R03135, R03136	17806 (Oxidized polyvinyl alcohol)
R06208, R00879	17643 (beta-D-Fructan)
R03167	17642 (Lipid)
R08144	17551 (Farnesol)
R03142	17956 (Monosaccharide 1-phosphate), 17505 (ADP-aldose)
R03143	17956 (Monosaccharide 1-phosphate), 17455 (NDP-aldose)
R0TCK	1337 (discovery supporting term)

continued on next page

8.2 Indistinctive Reactions

continued from previous page	
Reaction Code	Substance without chemical formula
R02042, R09474	17425 (Wax ester)
R03038, R08779, R08780	17421 (tRNA(Ala))
R03646, R03862	17420 (tRNA(Arg))
R03647, R03648	17419 (tRNA(Asn))
R05577	17418 (tRNA(Asp))
R03650, R08576	17417 (tRNA(Cys))
R03651, R03652	17416 (tRNA(Gln))
R04109, R05578	17415 (tRNA(Glu))
R03654, R08776, R08777, R08778	17414 (tRNA(Gly))
R03655	17413 (tRNA(His))
R03656	17412 (tRNA(Ile))
R03657	17411 (tRNA(Leu))
R03658	17410 (tRNA(Lys))
R03659, R04773	17409 (tRNA(Met))
R03660	17408 (tRNA(Phe))
R03661	17407 (tRNA(Pro))
R03662	17406 (tRNA(Ser))
R03663	17405 (tRNA(Thr))
R03664	17404 (tRNA(Trp))
R03665	17403 (tRNA(Val))
R02460	17391 (Bacitracin)
R03203	17898 (11beta-Hydroxysteroid), 17110 (11-Oxosteroid)
R02826	16899 (Aliphatic amide)
R02824	18214 (Insulin), 16865 (Reduced insulin)
R00076, R00077	16850 (Phosphorylase a), 16849 (Phosphorylase b)
R02903	16746 (Phosphorhodopsin)
R02130	18908 (Hydrogen-acceptor), 18905 (Hydrogen-donor), 18551 (Steroid), 16702 (21-Hydroxysteroid)
R03841	16589 (D-Hexose phosphate)
R04015	16581 (Ferricytochrome c3), 16579 (Ferrocytochrome c3)
R00101, R00109	16580 (Ferrileghemoglobin), 16578 (Ferroleghemoglobin)
R03632	17446 (Protamine), 16544 (O-Phosphoprotamine)
R03615	17472 (Flavonoid), 16498 (3'-Hydroxyflavonoid)
R02899	16439 (O-Sinapoylglucarate)
R05186	16532 (Reduced flavodoxin), 16436 (Oxidized flavodoxin)
R00034	16248 (Cyclobutadipyrimidine)

continued on next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page	
Reaction Code	Substance without chemical formula
R03744	17209 (tau-Protein), 16207 (O-Phospho-tau-protein)
R01348, R01840, R01842, R02351, R02501, R02503, R03087, R03629, R03697, R04121, R04122, R04759, R04761, R05259, R08053, R08054, R08055, R08066, R08068, R08225, R08257, R08264, R08265, R08551, R08785, R08840, R08841	16312 (Reduced flavoprotein), 16205 (Oxidized flavoprotein)
R03817, R08409	16311 (Reduced plastocyanin), 16204 (Oxidized plastocyanin)
R09542	18802 (Reduced ferredoxin), 18801 (Oxidized ferredoxin), 18733 (Photon), 16311 (Reduced plastocyanin), 16204 (Oxidized plastocyanin)
R04123	16308 (Transferrin[Fe(II)] ₂), 16190 (Transferrin[Fe(III)] ₂)
R04156	16187 (Glycogen-synthase D), 16186 (Glycogen-synthase I)
R04176	16136 (ADP-D-ribosyl-acceptor)
R03818, R10047, R10048	17041 (Putidaredoxin), 16102 (Oxidized putidaredoxin)
R05193	18538 (Calmodulin), 18432 (Ubiquitin), 16082 ((Ubiquitin)n-calmodulin)
R02131	18908 (Hydrogen-acceptor), 18905 (Hydrogen-donor), 18551 (Steroid), 16081 (17alpha-Hydroxysteroid)
R01407	18765 (Acid), 16056 (Acyl-protein thioester)
R01952	18603 (Glycoprotein), 16031 (N-Palmitoylglycoprotein)
R03861	15952 (O-Sinapoylglucarolactone)
R04039	16525 (Xanthine oxidase), 15942 ([Xanthine : NAD oxidoreductase])
R02432	18422 (Gentamicin), 15935 (2''-Nucleotidylgentamicin)
R03711	17284 (Actinomycin), 15908 (Actinomycinic monolactone)
R03876	18432 (Ubiquitin), 15882 (Protein N-ubiquityllysine)
R01623, R01625	15835 (Apo-[acyl-carrier-protein])
R01810	15825 (N-Acetyl-O-acetylneuraminate)
continued on next page	

8.2 Indistinctive Reactions

continued from previous page	
Reaction Code	Substance without chemical formula
R03743	17210 (beta-Lactam), 15751 (Substituted beta-amino acid)
R01994	18593 (Lipopolysaccharide), 15716 (D-Glucosyllipopolysaccharide)
R03981	16672 (Glycosaminoglycan), 15656 (D-Galactosylglycosaminoglycan)
R04146	16245 (D-Galactosaminoglycan), 15604 (N-Acetyl-D-galactosaminoglycan)
R02418	18432 (Ubiquitin), 15551 (Ubiquitin C-terminal thiolester)
R03768	17166 (Heteropolysaccharide), 15547 (2-alpha-D-Mannosyl-heteroglycan)
R03769	17166 (Heteropolysaccharide), 15546 (3-alpha-D-Mannosyl-heteroglycan)
R03923	16824 ('Activated' tRNA), 15495 (tRNA containing a thionucleotide)
R04149	16234 (Glycoprotein inositol), 15477 (Glycoprotein phosphatidylinositol)
R04275	15875 (Tyrosine-3-monooxygenase), 15461 (Phospho-[tyrosine-3-monooxygenase])
R01995	18593 (Lipopolysaccharide), 15423 (alpha-D-Glucosyllipopolysaccharide)
R04462	15352 (2-Hexadecenoyl-[acyl-carrier protein])
R02183	15334 (N-Acetyl-D-galactosaminyl-polypeptide)
R04358	15641 ([RNA polymerase]), 15329 (Phospho-[DNA-directed RNA polymerase])
R01997	18593 (Lipopolysaccharide), 15276 (3-alpha-D-Galactosyl-[lipopolysaccharide glucose])
R01996	18593 (Lipopolysaccharide), 15223 (N-Acetyl-D-glucosaminyllipopolysaccharide)
R00392, R00611, R01178, R01565, R01588, R02488, R02511, R03169, R04096, R04432, R04433, R05584, R10074	15434 (Electron-transferring flavoprotein), 15198 (Reduced electron-transferring flavoprotein)
R04511	15199 (Membrane-derived-oligosaccharide D-glucose), 14998 (Membrane-derived-oligosaccharide 6-(glycerophospho)-D-glucose)
R04589	15047 ([[Hydroxymethylglutaryl-CoA reductase (NADPH)]kinase]),

continued on next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page	
Reaction Code	Substance without chemical formula
R04588	14992 (Dephospho-[[hydroxymethylglutaryl-CoA reductase (NADPH)]kinase]) 15049 (Citrate-oxaloacetate-lyase((pro-3S)-CH ₂ COO- _i acetate)), 14982 (Deacetyl-[citrate-oxaloacetate-lyase((pro-3S)-CH ₂ COO- _i acetate)])
R00422, R01120, R01537, R04289	14872 (Glycolipid)
R00423	14837 (Mucopolysaccharide)
R04710	14784 (Dihydroflavodoxin), 14781 (Flavodoxin semiquinone)
R03986, R08845	17956 (Monosaccharide 1-phosphate), 14763 (UDP-sugar)
R04698	14791 (Ferrocytochrome b-561), 14711 (Ferricytochrome b-561)
R00083	14707 (Semiquinone)
R00785, R05751, R09480	14668 (Oxidized azurin), 14667 (Reduced azurin)
R00067, R00153, R02802, R04782, R05398, R05482, R05505, R05545, R06404, R07687, R08862, R08873, R08875, R09294, R09317, R09799, R10150, R10202	14666 (Electron)
R00002	18802 (Reduced ferredoxin), 18801 (Oxidized ferredoxin), 14666 (Electron)
R04299	14649 (Aminosugars)
R00657	14544 (Aminoacyl-L-methionine)
R04924	14438 (Lipophosphoglycan)
R04939, R05657, R09180	14425 (Methyl group acceptor), 14424 (Methyl-acceptor)
R04124	16308 (Transferrin[Fe(II)] ₂), 14356 (Apotransferrin)
R04155	16190 (Transferrin[Fe(III)] ₂), 14356 (Apotransferrin)
R03682	17356 (Hemoglobin), 14355 (Oxyhemoglobin)
R00312	14353 (Cytochrome a)
R00313	14352 (Catalase)
R00314	14351 (Peroxidase)
R05010	14275 ((alpha-D-Mannosyl)9-beta-D-mannosyl-diacetylchitobiosyldiphosphodolichol)
R02663, R03175, R04098	14188 (Branched chain fatty acid)
R01474, R03079,	14176 (Pentosan)

continued on next page

8.2 Indistinctive Reactions

continued from previous page	
Reaction Code	Substance without chemical formula
R04937, R04938	
R01400	14133 (Cyclic amidines)
R06138	14132 (Amidines)
R02876	14114 (Hopanoid)
R03201	14113 (Triterpenoid)
R05102	14092 (Cytochrome P-450 oxidized form), 14091 (Cytochrome P-450 reduced form)
R04644, R04645, R05106	14072 (Ceramidepentasaccharide)
R03683, R03684	17356 (Hemoglobin), 13957 (Globin)
R04977	14354 (Myoglobin), 13957 (Globin)
R00792	13956 (Ferrocyclochrome b), 13955 (Ferricycyclochrome b)
R05163	13879 (Hydroxycinnamoyl-CoA), 13878 (Anthocyanidin 3-glucoside-5-hydroxycinnamoylglucoside)
R01705	13832 (Palmitoyl-protein)
R07292	13511 (Feruloyl-polysaccharide)
R06054	18593 (Lipopolysaccharide), 13508 (3-Deoxyoctulosonyl-lipopolysaccharide)
R05741	12822 (Dihydrofurano derivative)
R05747	12821 (Glutaredoxin), 12820 (Glutaredoxin disulfide)
R05793	12815 (Cutin), 12814 (Cutin monomer)
R05744, R07643, R10011	12809 (Products of ATP breakdown)
R09422	14357 (Apoferritin), 1090 (Ferritin)
R09469	1072 (Diketone)
R09428	1057 (Acceptor), 1056 (Acceptor beta-D-glucuronoside)
R00606	1032 (Amicyanin), 1031 (Reduced amicyanin)
R10152	4985 (Quinone), 4984 (Hydroquinone), 1011 (Polysulfide)

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

8.3 Unbalanced Reactions

Here is the list of unbalanced reactions excluding those detected to be transmutational:

Table 8.3: Unbalanced reactions. “No hint” means, that the corresponding KEGG site not clearly points out the issue

reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R00001	35×H, 105×O, 35×P			✓
R00017	2×H			✓
R00025	H			✓
R00041	4×C, 8×H			✓
R00092	2×H			✓
R00102	H			✓
R00103	H			✓
R00119	H			✓
R00135	5×C, 7×H, N, O			✓
R00137	H			✓
R00263	20×C, 26×H, 4×N, 11×O	✓		
R00344	H			✓
R00379	5×C, 8×H, 5×O, P, R			✓
R00380	24×C, 37×H, 3×N, 24×O, 5×P, 3×R			✓
R00381	35×C, 56×H, 35×O, 7×P, 7×R			✓
R00382	35×C, 57×H, 35×O, 7×P, 7×R			✓
R00383	H			✓
R00384	2×C, 4×H			✓
R00385	2×C, 4×H			✓
R00387	2×C, 4×H			✓
R00388	2×C, 4×H			✓
R00390	2×C, 4×H			✓
R00393	H			✓
R00444	5×C, 8×H, 6×O, P, R			✓
R00459	3×H, N	✓		
R00538	H			✓
R00539	H			✓
R00540	H			✓
R00542	H			✓
R00543	H			✓
R00545	H			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page

reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R00546	H			✓
R00547	H			✓
R00555	H			✓
R00575	H	✓		
R00630	H			✓
R00634	H			✓
R00635	H			✓
R00649	H			✓
R00698	2×H, O		✓	
R00731	2×H			✓
R00742	H	✓		
R00774	H			✓
R00864	H			✓
R00887	6×C, 10×H, 5×O			✓
R00900	H			✓
R00915	4×C, 2×H, 2×O		✓	
R00916	2×H, 2×O		✓	
R00991	3×C, 2×H, 2×O	✓	✓	
R00993	O			✓
R01027	H			✓
R01028	H			✓
R01029	H			✓
R01032	H		✓	
R01235	C, 3×H, R			✓
R01260	H			✓
R01263	H			✓
R01273	H			✓
R01309	H	✓		
R01310	H			✓
R01312	H			✓
R01315	H			✓
R01316	H			✓
R01317	19×C, 32×H, R			✓
R01318	H			✓
R01319	H			✓
R01332	2×H, O	✓		

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R01350	2×C, 4×H, O, 2×R			✓
R01369	H			✓
R01408	H			✓
R01409	R			✓
R01410	H	✓	✓	
R01427	O		✓	
R01493	H			✓
R01553	H			✓
R01578	12×C, 14×H, 5×N, 8×O, 5×R		✓	
R01581	H			✓
R01637	O		✓	
R01650	H	✓	✓	
R01675	45×C, 69×H, 5×N, 51×O, 9×P, 7×R			✓
R01722	H			✓
R01724	H			✓
R01767	H			✓
R01798	19×C, 31×H, R			✓
R01859	H	✓		
R01861	H			✓
R01862	H			✓
R01890	H			✓
R01891	H			✓
R01913	C, 8×H, 5×O		✓	
R01920	H			✓
R01921	H			✓
R01929	H			✓
R01930	H			✓
R01931	2×H			✓
R01998	2×C, 4×H			✓
R02008	2×H		✓	
R02040	3×C, 6×H			✓
R02041	2×C, 4×H			✓
R02114	H			✓
R02116	13×C, 22×H		✓	
R02122	12×O, 4×S			✓
R02129	2×H		✓	

continued on the next page

8.3 Unbalanced Reactions

continued from previous page

reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R02170	H			✓
R02180	12×O, 4×S			✓
R02181	12×O, 4×S			✓
R02182	H			✓
R02184	H, 3×O, P			✓
R02185	H, 3×O, P			✓
R02186	H, 3×O, P			✓
R02187	H, 3×O, P			✓
R02188	H, 3×O, P			✓
R02189	H, 3×O, P			✓
R02192	H			✓
R02294	H			✓
R02300	H			✓
R02322	H			✓
R02323	H			✓
R02324	H			✓
R02409	O		✓	
R02420	12×C, 13×H, 6×N, 6×O, 5×R			✓
R02572	H			✓
R02573	H			✓
R02605	H			✓
R02617	2×C, 2×H			✓
R02676	H			✓
R02682	2×C, 4×H			✓
R02694	2×C, 2×H			✓
R02718	12×C, 20×H, 4×N, 4×O		✓	
R02744	H			✓
R02745	C, 2×H			✓
R02760	2×C, 2×H			✓
R02768	2×C, 4×H			✓
R02797	O			✓
R02816	O		✓	
R02829	C, H, R			✓
R02846	H			✓
R02862	H			✓
R02869	H			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R02895	2×H		✓	
R03042	H, 3×O, P			✓
R03105	H			✓
R03110	H			✓
R03129	6×C, 8×H, 6×O			✓
R03132	H			✓
R03223	H			✓
R03231	H			✓
R03348	H			✓
R03360	H			✓
R03376	2×H, O			✓
R03415	C, 2×H			✓
R03447	H			✓
R03451	H			✓
R03467	2×H	✓		
R03494	H			✓
R03553	3×O, S			✓
R03580	2×H	✓	✓	
R03666	5×H, N	✓		
R03706	6×C, 8×H, 6×O			✓
R03720	O			✓
R03722	H			✓
R03756	2×C, 2×H			✓
R03765	3×C, 6×H	✓	✓	
R03807	2×H, O		✓	
R03813	5×C, 4×H, 5×N, R			✓
R03832	H			✓
R03838	2×C, 2×H			✓
R03872	O		✓	
R03873	2×H, O		✓	
R03909	H			✓
R03911	2×C, 2×H, O		✓	
R03948	H			✓
R03950	4×H		✓	
R03951	3×H, 2×N, 3×O	✓	✓	
R03995	2×H		✓	

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R04003	H			✓
R04041	24×C, 40×H, 20×O			✓
R04042	3×C, 7×H, 5×O, P			✓
R04044	C, 2×H			✓
R04074	H			✓
R04138	H			✓
R04227	H			✓
R04241	5×C, 7×H, N, 3×O			✓
R04249	O			✓
R04250	C, 2×H		✓	
R04251	O		✓	
R04252	24×C, 38×H, 4×O		✓	
R04257	O			✓
R04276	H			✓
R04283	2×H, 2×O	✓	✓	
R04311	11×C, 21×H			✓
R04313	23×C, 24×H, 11×N, 12×O, 10×R	✓		
R04319	2×C, 4×H			✓
R04321	H			✓
R04326	H			✓
R04332	4×C, 8×H			✓
R04369	12×C, 20×H, 4×N, 4×O			✓
R04375	H			✓
R04384	H			✓
R04386	H	✓		
R04436	H			✓
R04456	H			✓
R04458	C, O		✓	
R04473	2×C, 4×H			✓
R04484	8×C, 8×H, 4×O			✓
R04505	H			✓
R04514	H			✓
R04524	O		✓	
R04534	H			✓
R04540	O			✓
R04541	O			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R04546	2×H, O	✓	✓	
R04616	2×H, O	✓	✓	
R04670	2×C, 5×H		✓	
R04685	8×C, 8×H, 4×N, 4×O, 4×R			✓
R04691	H			✓
R04692	H			✓
R04705	H			✓
R04707	H			✓
R04711	H			✓
R04713	H			✓
R04714	H			✓
R04715	H			✓
R04716	H			✓
R04717	H			✓
R04739	H			✓
R04763	4×H	✓	✓	
R04771	H		✓	
R04772	H			✓
R04775	O		✓	
R04795	C, 6×H, O	✓		
R04806	H			✓
R04807	H			✓
R04860	O		✓	
R04864	2×C, 4×H			✓
R04867	H			✓
R04869	23×C, 25×H, 11×N, 12×O, 10×R			✓
R04877	H			✓
R04879	C, 2×H, 2×O			✓
R04885	2×H			✓
R04899	O		✓	
R04906	2×C, 3×H, N, O		✓	
R04923	3×C, 7×H	✓	✓	
R04925	C, 2×H		✓	
R04932	2×H			✓
R04933	2×O		✓	
R04934	3×C, 4×H, O		✓	

continued on the next page

8.3 Unbalanced Reactions

continued from previous page

reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R04943	2×H			✓
R04980	H			✓
R04989	H			✓
R05001	O			✓
R05077	2×H			✓
R05079	4×H			✓
R05088	2×H, O	✓	✓	
R05089	2×H, O		✓	
R05090	7×C, 12×H, 6×O	✓	✓	
R05118	8×C, 12×H, 4×O			✓
R05131	C, 2×H		✓	
R05197	5×C, 7×H, N, 3×O			✓
R05209	H			✓
R05220	H			✓
R05223	H			✓
R05252	2×H			✓
R05265	H			✓
R05270	O			✓
R05325	2×H, O			✓
R05335	H			✓
R05344	2×H, 2×O	✓	✓	
R05416	H			✓
R05419	2×H, O			✓
R05430	4×H			✓
R05431	4×H			✓
R05432	4×H			✓
R05433	4×H			✓
R05437	4×H			✓
R05438	4×H			✓
R05470	3×H, N, O			✓
R05472	7×C, 4×Cl, 4×H, 2×O	✓		
R05474	C, 2×Cl, 2×H	✓		
R05476	H			✓
R05479	C, 2×H, O			✓
R05480	C, 2×Cl, 2×H, O	✓		
R05481	4×H			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R05492	2×H			✓
R05497	H			✓
R05537	4×H, 2×O			✓
R05541	2×H			✓
R05542	2×H			✓
R05556	5×C, 8×H	✓		
R05617	10×C, 18×H			✓
R05669	2×H, O			✓
R05670	3×H, N, O			✓
R05671	2×H, O		✓	
R05755	3×H			✓
R05780	H	✓		
R05794	H			✓
R05796	40×C, 68×H, 9×O			✓
R05797	20×C, 28×H, 9×O			✓
R01331 / R05816	6×C, 10×H, 5×O			✓
R05829	2×H			✓
R05840	C, 6×H, 4×O			✓
R05846	H			✓
R05849	4×H		✓	
R05852	3×H		✓	
R05853	2×H, O		✓	
R05854	2×H, O		✓	
R05859	2×H, O		✓	
R05860	2×H, O		✓	
R05867	O		✓	
R05868	2×O		✓	
R05869	2×H, O		✓	
R05871	O		✓	
R05872	O		✓	
R05873	H			✓
R05885	H			✓
R05886	C, 2×H, 2×O		✓	
R05887	C, 4×H, O			✓
R05888	C, 2×H		✓	
R05889	C, 4×H, O			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R05890	C, 2×H		✓	
R05891	6×C, 6×H, 8×O	✓	✓	
R05892	10×C, 13×H, N, O		✓	
R05893	9×C, 11×H, N, O		✓	
R05894	10×C, 10×H, 2×N		✓	
R05895	2×H, O		✓	
R05896	O		✓	
R05897	2×H, O		✓	
R05898	2×H, O		✓	
R05899	O		✓	
R05900	2×C, O		✓	
R04636 / R05901	30×C, 51×H, 3×N, 21×O			✓
R04575 / R05912	18×C, 29×H, 3×N, 13×O, 2×R			✓
R04590 / R05913	18×C, 29×H, 3×N, 13×O, 2×R			✓
R04607 / R05928	6×C, 8×H, 6×O			✓
R05933	3×C, 5×H, N, 2×O			✓
R05934	3×C, 5×H, N, 2×O			✓
R03116 / R06018	6×C, 10×H, 5×O			✓
R03118 / R06020	6×C, 10×H, 5×O			✓
R02888 / R06022	6×C, 10×H, 5×O			✓
R02889 / R06023	6×C, 10×H, 5×O			✓
R02335 / R06028	8×C, 13×H, N, 5×O			✓
R02890 / R06030	6×C, 10×H, 5×O			✓
R03996 / R06045	6×C, 10×H, 5×O			✓
R02421 / R06049	6×C, 10×H, 5×O			✓
R01821 / R06050	6×C, 10×H, 5×O			✓
R00292 / R06051	6×C, 10×H, 5×O			✓
R01823 / R06052	6×C, 10×H, 5×O			✓
R04194 / R06059	6×C, 10×H, 5×O			✓
R06063	H			✓
R06064	H			✓
R02120 / R06066	6×C, 10×H, 5×O			✓
R05327 / R06068	5×O			✓
R05196 / R06069	60×C, 100×H, 50×O			✓
R04343 / R06072	6×C, 8×H, 6×O			✓
R06046 / R06074	78×C, 131×H, 70×O			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R02887 / R06077	6×C, 10×H, 5×O			✓
R03115 / R06078	6×C, 10×H, 5×O			✓
R05140 / R06079	6×C, 10×H, 5×O			✓
R01718 / R06080	6×C, 12×H, 6×O			✓
R01206 / R06081	8×C, 13×H, N, 5×O			✓
R03928 / R06083	5×C, 8×H, 4×O			✓
R06131	O			✓
R06132	C, 2×H			✓
R06134	H			✓
R06149	6×C, 10×H, 5×O			✓
R06150	6×C, 10×H, 5×O			✓
R06151	6×C, 10×H, 5×O			✓
R02109 / R06158	6×C, 9×H, 4×O			✓
R02112 / R06159	18×C, 29×H, 15×O			✓
R03122 / R06160	12×C, 18×H, 9×O			✓
R02833 / R06175	42×C, 77×H, 7×N, 24×O			✓
R04519 / R06177	40×C, 64×H, 8×N, 21×O			✓
R06178 / R06179	39×C, 64×H, 8×N, 19×O			✓
R02716 / R06181	24×C, 40×H, 20×O			✓
R02717 / R06182	32×C, 52×H, 4×N, 20×O			✓
R02111 / R06185	6×C, 9×H, 4×O			✓
R02110 / R06186	H, O			✓
R01790 / R01791 / R06199	6×C, 10×H, 5×O			✓
R02886 / R06200	12×C, 20×H, 10×O			✓
R01982 / R04320 / R06201	6×C, 8×H, 6×O			✓
R01105 / R06202	6×C, 10×H, 5×O			✓
R01433 / R06203	5×C, 8×H, 4×O			✓
R00308 / R06204	6×C, 10×H, 5×O			✓
R01762 / R06205	10×C, 18×H, 9×O			✓
R05624 / R06206	42×C, 70×H, 35×O			✓
R02108 / R06209	90×C, 149×H, 75×O			✓
R00506 / R06210	6×C, 10×H, 5×O			✓
R00890 / R06213	6×C, 10×H, 5×O			✓
R03078 / R06231	5×C, 8×H, 4×O			✓
R04083 / R06235	5×C, 11×H, 7×O, P			✓
R02360 / R06239	12×C, 16×H, 12×O			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R02361 / R06240	12×C, 14×H, 11×O			✓
R02333 / R06241	54×C, 103×H, 9×N, 34×O			✓
R02465 / R06242	5×O			✓
R01824 / R06243	48×C, 78×H, 39×O			✓
R02362 / R06250	95×C, 128×H, 89×O			✓
R05191 / R06278	100×C, 140×H, 90×O			✓
R06287	20×C, 38×H		✓	
R06288	2×H		✓	
R06293	H	✓		
R06297	H	✓		
R06316	3×O		✓	
R06318	2×C, 7×O	✓	✓	
R06319	2×H, 5×O	✓	✓	
R06324	O			✓
R06325	2×H			✓
R06326	2×H, O		✓	
R06327	2×H, O		✓	
R06328	O		✓	
R06329	O		✓	
R06330	O		✓	
R06331	O		✓	
R06332	O		✓	
R06333	O			✓
R06334	O			✓
R06339	O		✓	
R06340	O		✓	
R06341	O			✓
R06342	2×H			✓
R06343	2×H			✓
R06344	2×H			✓
R06345	2×H			✓
R06346	2×H		✓	
R06347	O		✓	
R06348	O		✓	
R06349	2×H		✓	
R06350	O		✓	

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R06355	2×H		✓	
R06356	2×H		✓	
R06357	H			✓
R06358	O		✓	
R06359	O		✓	
R06363	C, O			✓
R06367	2×H, O			✓
R06368	2×H, O			✓
R06374	2×H		✓	
R06375	2×H		✓	
R06394	2×H		✓	
R06395	2×H		✓	
R06396	2×H, O			✓
R06405	H		✓	
R06432	H			✓
R06435	H			✓
R06438	2×H			✓
R06439	2×H			✓
R06441	2×H			✓
R06445	O			✓
R06446	2×H			✓
R06462	H			✓
R06463	H			✓
R06465	H			✓
R06467	H			✓
R06468	H			✓
R06470	H			✓
R06500	18×C, 28×H, 15×O	✓		
R06503	2×C, 2×H, O			✓
R06504	4×C, 4×H, 2×O			✓
R06505	5×C, 6×H, 2×O			✓
R06515	2×H, O			✓
R06548	C, 2×H			✓
R06549	C, 2×H			✓
R06550	2×C, 4×H			✓
R06553	2×H			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page

reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R06554	2×H			✓
R06565	O	✓		
R06579	O			✓
R06580	O			✓
R06584	O			✓
R06586	3×H, N, O			✓
R06597	2×H		✓	
R06598	O		✓	
R06599	2×H		✓	
R06600	2×C, 2×H, 2×O		✓	
R06605	H		✓	
R06617	2×H			✓
R06618	O			✓
R06619	O			✓
R06621	H			✓
R06630	2×H		✓	
R06635	21×C, 36×H, 7×N, 16×O, 3×P, S			✓
R06637	21×C, 36×H, 7×N, 16×O, 3×P, S			✓
R06644	21×C, 36×H, 7×N, 16×O, 3×P, S			✓
R06645	21×C, 36×H, 7×N, 16×O, 3×P, S			✓
R06651	C, 2×H			✓
R06652	O			✓
R06653	O			✓
R06654	C, 2×H			✓
R06655	2×H, 2×O			✓
R06656	O			✓
R06661	O			✓
R06670	6×C, 10×H, 3×O		✓	
R06671	C, 2×H	✓		
R06673	3×H, N			✓
R06674	4×C, 8×H, 5×O	✓		
R06678	2×H			✓
R06689	2×H			✓
R06692	2×H			✓
R06694	2×H			✓
R06695	2×H			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R06697	6×H			✓
R06701	C, 2×H			✓
R06702	C, 2×H			✓
R06706	C, 2×H			✓
R06707	C, 2×O			✓
R06709	C, 2×H			✓
R06710	C, 2×O			✓
R06711	C, 2×O			✓
R06712	C, 4×H, 3×O			✓
R06713	C, 2×H			✓
R06714	C, 4×H, 3×O			✓
R06725	2×H, O			✓
R06726	2×C, 14×H, 2×N, O			✓
R06727	5×C, 5×H, 2×O			✓
R06731	4×H, 7×O, 2×P			✓
R06733	2×H	✓		
R06737	C, 2×H, 2×O	✓		
R06738	C, 2×H			✓
R06743	5×C, 3×H, N			✓
R06748	2×H			✓
R06752	2×H			✓
R06753	2×H			✓
R06755	O			✓
R06760	5×C, 8×H			✓
R06762	2×C, 2×H, 2×O		✓	
R06763	O		✓	
R06764	O			✓
R06776	2×H, O			✓
R06780	O		✓	
R06781	O		✓	
R06790	O		✓	
R06791	O		✓	
R06798	15×C, 16×H, 8×O			✓
R06822	6×C, 10×H, 5×O			✓
R06823	6×C, 10×H, 4×O			✓
R06825	6×C, 10×H, 5×O			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R06837	H, 3×O, P		✓	
R06849	Cl, H		✓	
R06850	Cl, H		✓	
R06852	2×H	✓		
R06853	2×H			✓
R06854	2×H			✓
R06855	2×H, O		✓	
R06858	2×H	✓	✓	
R06861	H			✓
R06862	H			✓
R06863	H			✓
R06865	O			✓
R06869	H			✓
R06876	2×H			✓
R06877	2×H			✓
R06878	2×H			✓
R06879	2×H			✓
R06880	12×C, 12×H, 2×O		✓	
R06881	O		✓	
R06882	2×H		✓	
R06887	2×H, O		✓	
R06889	2×H, 2×O	✓		
R06898	4×H		✓	
R06899	6×H		✓	
R06900	2×C, 2×H, 2×O		✓	
R06920	2×H			✓
R06929	3×C, 2×H, 3×O		✓	
R06938	3×C, 4×H, O		✓	
R06955	O			✓
R06956	2×H			✓
R06958	2×H			✓
R06959	2×H, O	✓		
R06961	2×H			✓
R06986	12×C, 20×H, 10×O			✓
R06991	O			✓
R06992	2×H			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R06993	4×H			✓
R06994	C, 2×H			✓
R06995	C			✓
R06996	2×H			✓
R06999	2×H			✓
R07005	5×C, 8×H, 2×N, 3×O	✓	✓	
R07006	5×C, 8×H, 2×N, 3×O	✓	✓	
R07008	5×C, 8×H, 2×N, 3×O	✓	✓	
R07017	2×H			✓
R07018	O		✓	
R07019	2×H		✓	
R07028	O		✓	
R07029	2×H		✓	
R07064	17×C, 32×H, R			✓
R07077	2×H		✓	
R07078	2×H		✓	
R07096	Cl, H			✓
R07103	Cl, H, O		✓	
R07117	H			✓
R07118	5×C, 8×H, 2×N, 3×O	✓		
R07119	4×H, 3×N, 2×O		✓	
R07120	2×H		✓	
R07123	Cl, H		✓	
R07125	H			✓
R07162	2×C, 4×H			✓
R07169	H			✓
R07177	2×H			✓
R07214	H			✓
R07232	45×C, 69×H, 5×N, 51×O, 9×P, 7×R			✓
R07233	45×C, 69×H, 5×N, 52×O, 9×P, 7×R			✓
R07234	45×C, 69×H, 5×N, 52×O, 9×P, 7×R			✓
R07241	H			✓
R07249	8×C, 8×H, 4×O			✓
R07255	H, R			✓
R07261	6×C, 10×H, 5×O			✓
R07266	O			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R07268	H			✓
R07282	5×C, 8×H, 6×O, P, R			✓
R07285	5×C, 8×H, 6×O, P, R			✓
R07287	H			✓
R07288	12×O, 4×S			✓
R07291	H			✓
R07293	H			✓
R07312	H			✓
R07314	C, 2×H			✓
R07326	C, 2×H			✓
R07327	C, H, R			✓
R07328	C, 2×H			✓
R07332	O			✓
R07333	O			✓
R07338	O			✓
R07344	O			✓
R07351	H			✓
R07352	H			✓
R07377	H			✓
R07379	H			✓
R07387	H			✓
R07389	H			✓
R05930 / R07397	8×C, 13×H, N, 5×O			✓
R07404	H			✓
R07412	2×H, O			✓
R07430	H			✓
R07436	O			✓
R07444	2×H			✓
R07445	H			✓
R07446	2×H		✓	
R07448	2×H			✓
R07449	2×H			✓
R07450	2×H			✓
R07451	2×H			✓
R07452	H			✓
R07453	H			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R07454	H			✓
R07466	2×H		✓	
R07468	2×H			✓
R07469	2×H			✓
R07471	O			✓
R07472	2×H			✓
R07473	2×H			✓
R07474	2×H		✓	
R07482	C, 2×H			✓
R07485	C, 2×H			✓
R07486	2×H			✓
R07487	2×H			✓
R07489	2×H			✓
R07490	C, 2×H			✓
R07491	2×H			✓
R07492	2×H			✓
R07496	C, 2×H	✓	✓	
R07508	C, 2×H			✓
R07510	4×H	✓	✓	
R07517	2×H			✓
R07523	2×H			✓
R07534	2×H			✓
R07540	2×H		✓	
R07541	4×H	✓	✓	
R07542	2×H	✓	✓	
R07544	6×C, 10×H, 5×O			✓
R07546	6×C, 10×H, 5×O			✓
R07549	2×H, O			✓
R07550	O			✓
R07551	2×H			✓
R07553	8×C, 14×H, 4×O	✓		
R07554	O			✓
R07556	O			✓
R07557	2×H, O			✓
R07560	8×H	✓	✓	
R07563	2×H, O			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page

reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R07564	2×H, O			✓
R07565	2×H, O			✓
R07566	2×H, O			✓
R07567	2×H, O			✓
R07571	2×H, O			✓
R07573	2×H, O			✓
R07577	2×H			✓
R07578	18×C, 32×H, 6×O			✓
R07579	O			✓
R07580	18×C, 32×H, 6×O			✓
R07581	18×C, 32×H, 6×O			✓
R07633	H			✓
R07634	H			✓
R07640	35×C, 56×H, 42×O, 7×P, 7×R			✓
R07641	H	✓		
R07655	6×C, 10×H, 5×O		✓	
R07656	15×C, 28×H, O		✓	
R07663	2×H		✓	
R07682	2×H		✓	
R07683	2×H		✓	
R07685	2×H	✓	✓	
R07686	2×H		✓	
R07690	3×C, O		✓	
R07692	2×H	✓	✓	
R07693	C, 2×H, 2×O		✓	
R07694	C, 2×H		✓	
R07695	3×C, 2×H, O		✓	
R07696	C, O		✓	
R07697	2×H		✓	
R07698	2×H		✓	
R07699	2×H		✓	
R07700	2×H		✓	
R07706	2×H		✓	
R07712	2×H		✓	
R07716	2×H, O		✓	
R07717	C, 2×H		✓	

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R07726	2×H, O		✓	
R07732	3×C, 2×H, 3×O			✓
R07733	6×C, 10×H, 5×O			✓
R07735	2×C, 4×H, O	✓	✓	
R07737	2×H			✓
R07738	2×H, O			✓
R07742	3×C, 2×H, 3×O			✓
R07743	6×C, 10×H, 5×O			✓
R07744	7×C, 10×H, O	✓	✓	
R07749	O		✓	
R07750	C, 2×H			✓
R07755	3×C, 2×H, 3×O			✓
R07756	6×C, 10×H, 5×O			✓
R07773	H			✓
R07781	H			✓
R07783	C, 2×O		✓	
R07784	O		✓	
R07785	Br, H, 3×O		✓	
R07788	2×H		✓	
R07789	2×H		✓	
R07790	2×H		✓	
R07803	3×C, 2×H		✓	
R07805	34×C, 56×H, 2×N, 34×O, 3×S			✓
R07810	3×O, S			✓
R07826	O		✓	
R07834	2×H, O			✓
R07841	O			✓
R07842	O			✓
R07843	6×C, 10×H, 5×O			✓
R07845	12×C, 20×H, 8×O			✓
R07847	C, 2×H, 2×O	✓	✓	
R07849	4×H			✓
R07857	4×H			✓
R07858	4×H			✓
R07859	17×C, 30×H, R			✓
R07860	17×C, 30×H, R			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R07862	2×H, 2×O			✓
R07900	15×C, 16×H, 7×O	✓	✓	
R07901	C, 2×H			✓
R07904	15×C, 16×H, 7×O		✓	
R07905	C, 2×H			✓
R07906	C, 2×H			✓
R07907	2×C, 4×H			✓
R07908	6×C, 10×H, 5×O			✓
R07909	6×C, 10×H, 5×O			✓
R07915	60×C, 64×H, 28×O	✓	✓	
R07917	H			✓
R07918	H			✓
R07924	9×C, 6×H, 3×O			✓
R07925	18×C, 12×H, 6×O			✓
R07940	C, 2×H, 2×O			✓
R07941	C, O			✓
R07943	C, 2×H			✓
R07945	O			✓
R07949	H	✓		
R07994	O			✓
R08000	2×H			✓
R08001	2×H			✓
R08004	C, 2×H			✓
R08011	O			✓
R08012	C, 2×H			✓
R08023	18×C, 24×H, 11×O	✓	✓	
R08026	C, 2×H, O	✓		
R08028	2×C, 2×H, O			✓
R08029	C, 2×H			✓
R08031	C, 2×H			✓
R08033	H, N, 2×O		✓	
R08034	2×H, O			✓
R08041	2×O		✓	
R08049	O		✓	
R08065	2×H			✓
R08067	2×H		✓	

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R08077	O			✓
R08123	C, 2×H, 2×O			✓
R08126	5×C, 8×H, 4×O			✓
R08129	6×C, 10×H, 5×O			✓
R08130	2×H			✓
R08131	2×H			✓
R08132	2×H, 2×O	✓		
R08133	O			✓
R08134	O			✓
R08135	O			✓
R08136	O			✓
R08137	2×H			✓
R08138	O			✓
R08139	O			✓
R08142	O			✓
R08143	O			✓
R08147	C, 2×H			✓
R08155	O			✓
R08156	O			✓
R08201	H			✓
R08211	H	✓		
R08212	H	✓		
R08217	O			✓
R08234	2×H		✓	
R08252	2×H, O		✓	
R08256	2×O	✓	✓	
R08268	C, 2×H		✓	
R08269	C, 2×H		✓	
R08272	C, 2×H		✓	
R08276	O		✓	
R08277	2×H		✓	
R08286	O		✓	
R08287	O		✓	
R08290	2×H		✓	
R08291	2×H		✓	
R08292	O		✓	

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R08293	2×C, 4×H		✓	
R08296	2×C, 4×H		✓	
R08297	2×C, 4×H		✓	
R08298	O		✓	
R08315	O		✓	
R08316	2×H		✓	
R08317	O		✓	
R08318	C, H, N, O		✓	
R08319	2×H		✓	
R08320	O		✓	
R08321	2×H		✓	
R08322	O		✓	
R08324	2×H		✓	
R08329	2×H		✓	
R08330	2×H		✓	
R08333	C, 2×H, O		✓	
R08334	2×H, O		✓	
R08335	2×H		✓	
R08336	2×H		✓	
R08337	2×H		✓	
R08338	2×H, O		✓	
R08340	C, 2×H	✓	✓	
R08341	C, 2×H		✓	
R08342	C, 2×H		✓	
R08343	C, 2×H		✓	
R08344	O		✓	
R08345	C, 2×H		✓	
R08359	H			✓
R08375	2×H		✓	
R08377	5×C, 6×H, 15×O	✓	✓	
R08378	2×C, 6×H, 7×O	✓	✓	
R08387	H			✓
R08389	H			✓
R08400	4×C, 8×H			✓
R08404	H			✓
R08405	H			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R08406	H			✓
R08407	H			✓
R08412	C, 2×H, 3×O	✓	✓	
R08413	5×C, 7×H, N		✓	
R08414	2×H		✓	
R08415	2×C, 2×H, O	✓	✓	
R08416	2×H, 2×O	✓	✓	
R08417	2×C, 5×H, N, O	✓	✓	
R08419	10×C, 12×H	✓	✓	
R08422	4×H, O	✓	✓	
R08423	C, 4×H, O	✓		
R08424	2×H		✓	
R08425	2×H		✓	
R08426	2×H		✓	
R08436	O	✓	✓	
R08437	C, 2×H		✓	
R08438	2×H, O	✓	✓	
R08439	2×H	✓	✓	
R08440	2×H	✓	✓	
R08442	H			✓
R08443	C, 2×H, 2×O	✓	✓	
R08444	3×C, 6×H, O	✓	✓	
R08445	3×C, 4×H, O	✓	✓	
R08447	4×C, 6×H, O	✓	✓	
R08448	2×H		✓	
R08450	2×H, O	✓	✓	
R08452	C, 2×H		✓	
R08453	2×C, 2×H, O		✓	
R08454	2×H		✓	
R08456	C, 2×H, O	✓	✓	
R08457	2×C	✓	✓	
R08458	C, H		✓	
R08459	2×C		✓	
R08460	2×H		✓	
R08463	O	✓		
R08464	2×H		✓	

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R08466	2×H		✓	
R08470	4×C, 8×H, 2×O	✓	✓	
R08471	6×C, 8×H	✓	✓	
R08472	5×C, 6×H	✓	✓	
R08473	2×C	✓	✓	
R08474	6×C, 10×H, O	✓	✓	
R08475	3×C, 8×H, O	✓	✓	
R08476	2×C, 4×H, 2×O	✓	✓	
R08477	2×H, O	✓	✓	
R08478	2×C, 4×H, 6×O	✓	✓	
R08479	O	✓	✓	
R08482	2×C, 2×H	✓	✓	
R08483	2×C, 2×H		✓	
R08485	2×H		✓	
R08486	2×H		✓	
R08487	C, 4×H, O		✓	
R08488	2×H, O	✓	✓	
R08489	O		✓	
R08490	6×C, 12×H, 4×O	✓	✓	
R08491	2×H	✓	✓	
R08495	6×C, 12×H, 7×O	✓	✓	
R08496	8×C, 12×H, 8×O	✓	✓	
R08497	C, O		✓	
R08498	2×H	✓	✓	
R08500	C, 4×H, O		✓	
R08513	4×C, 6×H, O	✓	✓	
R08564	2×H	✓	✓	
R08577	5×C, 4×H, 5×N, R			✓
R08585	2×H, O		✓	
R08588	2×H		✓	
R08589	3×C, 6×H, 3×O		✓	
R08594	3×H, N, O		✓	
R08595	2×C, 4×H	✓	✓	
R08596	30×C, 48×H, 2×O	✓	✓	
R08598	2×H		✓	
R08604	H			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R08605	2×H		✓	
R08606	H, N		✓	
R08612	30×C, 50×H, 25×O			✓
R06207 / R08614	6×C, 10×H, 5×O			✓
R08621	2×H		✓	
R08625	2×H		✓	
R08629	2×H		✓	
R08636	2×H		✓	
R08642	2×H		✓	
R08646	2×H		✓	
R08682	2×H		✓	
R08690	C, 2×H	✓	✓	
R08692	6×C, 8×H, 6×O			✓
R08694	36×C, 47×H, 36×O			✓
R08699	2×H		✓	
R08700	2×H	✓	✓	
R08716	2×H		✓	
R08758	2×H			✓
R08759	2×H			✓
R08768	H			✓
R08770	O			✓
R08783	2×H			✓
R08786	6×C, 10×H, 6×O	✓	✓	
R08789	H			✓
R08791	H			✓
R08792	C, 2×H	✓	✓	
R08798	4×H	✓	✓	
R08799	4×H	✓	✓	
R08800	4×H	✓	✓	
R08801	O		✓	
R08802	O		✓	
R08811	C, 2×H, O		✓	
R08812	C, 2×H, O		✓	
R08834	H			✓
R08838	H			✓
R08843	2×H		✓	

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R08847	O		✓	
R08865	2×H		✓	
R08866	O			✓
R08881	C, 2×H	✓		
R08888	C, 2×H, 2×O	✓	✓	
R08927	2×H		✓	
R08928	3×H, N, O		✓	
R08929	2×C, 4×H	✓	✓	
R08930	O		✓	
R08936	H			✓
R08940	2×C, 4×H			✓
R08969	H			✓
R08984	3×O		✓	
R08985	3×C, 6×H		✓	
R08986	3×O		✓	
R08989	5×C, 4×H, 2×O	✓	✓	
R08990	O		✓	
R08994	2×H		✓	
R08995	2×C, 4×H, 2×O		✓	
R09041	C, O			✓
R09043	6×H, O			✓
R09044	O			✓
R09045	2×H, O			✓
R09046	2×H			✓
R09050	O			✓
R09052	2×H			✓
R09055	2×H			✓
R09056	2×H		✓	
R09061	2×C, 2×H, 2×O		✓	
R09066	O			✓
R09083	4×C, 2×H, 2×N, 3×O		✓	
R09088	2×H, O			✓
R09091	2×H, O		✓	
R09092	2×H, O			✓
R09096	H			✓
R09100	3×C, 7×H, 5×O, P		✓	

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R09102	O			✓
R09103	2×H			✓
R09104	O			✓
R09109	2×H		✓	
R09110	O		✓	
R09113	2×H			✓
R09114	O		✓	
R09118	2×H, O		✓	
R09143	Cl, H, O	✓		
R09149	2×H			✓
R09150	O			✓
R09153	2×H, O, S			✓
R09154	2×H			✓
R09155	2×H			✓
R09161	2×H			✓
R09166	2×H			✓
R09167	2×H			✓
R09169	2×H			✓
R09171	2×H			✓
R09173	2×H			✓
R09174	2×H			✓
R09176	2×H			✓
R09178	2×H			✓
R09182	2×H		✓	
R09188	4×H, 2×O			✓
R09197	3×H, N, O			✓
R09198	O			✓
R09199	3×H, N, O			✓
R09202	3×C, 2×O		✓	
R09203	3×C, 2×O		✓	
R09205	2×H			✓
R09210	O			✓
R09212	2×H			✓
R09216	2×H			✓
R09218	2×H, 2×O			✓
R09239	2×H			✓

continued on the next page

8.3 Unbalanced Reactions

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R09252	2×H		✓	
R09314	2×H			✓
R09330	O			✓
R09333	6×C, 10×H, 2×O			✓
R09341	4×H			✓
R09343	4×H			✓
R09344	6×H, 2×O	✓	✓	
R09346	O			✓
R09347	6×C, 10×H, 3×O			✓
R09348	6×C, 10×H, 3×O	✓		
R09349	6×C, 10×H, 2×O	✓		
R09354	2×H			✓
R09356	10×C, 18×H, 4×O			✓
R09357	O			✓
R09359	2×R			✓
R09360	6×C, 10×H, 3×O			✓
R09361	6×C, 10×H, 3×O	✓		
R09362	6×C, 10×H, 2×O	✓		
R09370	H			✓
R09380	2×C, 4×H			✓
R09381	2×C, 4×H			✓
R09382	5×C, 8×H, 6×O, P, R	✓		
R09383	5×C, 8×H, 6×O, P, R			✓
R09412	2×H			✓
R09413	2×H			✓
R09414	2×H			✓
R09415	2×H			✓
R09429	O			✓
R09430	O			✓
R09437	O			✓
R09438	2×H, O			✓
R09439	O			✓
R09440	2×H, O			✓
R09448	2×H			✓
R09455	2×H			✓
R09456	2×H			✓

continued on the next page

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R09457	O			✓
R09458	O			✓
R09459	O			✓
R09466	2×C, 4×H			✓
R09471	4×H	✓	✓	
R09538	H			✓
R09590	16×C, 20×H, 4×N, 12×O			✓
R09597	2×H, O			✓
R09637	H			✓
R09639	H			✓
R09646	H			✓
R09739	H			✓
R09761	O			✓
R09762	O			✓
R09795	H			✓
R09809	H, O		✓	
R09845	6×C, 6×H, 3×N, 3×O, 3×R			✓
R09848	2×H		✓	
R09852	C, 2×H			✓
R09855	O			✓
R09919	8×C, 18×H	✓		
R09930	H	✓		
R09982	O			✓
R09994	12×C, 20×H, 10×O			✓
R09995	48×C, 78×H, 39×O	✓		
R10000	H			✓
R10001	H			✓
R10012	H			✓
R10014	H			✓
R10015	H			✓
R10016	H			✓
R10046	H			✓
R10123	5×C, 8×H, 4×O			✓
R10151	H			✓
R10163	H			✓
R10164	H			✓

continued on the next page

8.4 Transmutations

continued from previous page				
reaction code	unbalanced elements	multi-step	incomplete or unclear reaction	⚡ no hint
R10191	H			✓
R10229	H			✓
R10230	H			✓

8.4 Transmutations

Table 8.4: “Transmutations” reactions. “No hint” means, that the corresponding KEGG site not clearly points out the issue

Reaction Code	multi-step	unclear reaction	⚡no hint⚡	elements appearing
R00152		✓		H, O
R00165		✓		P
R00166		✓		P
R00167		✓		P
R00168		✓		P
R00169		✓		P
R00170			✓	P
R00172		✓		P, Se
R00375			✓	N
R00376			✓	N
R00377			✓	N
R00378			✓	N
R00435			✓	N
R00437			✓	N
R00438			✓	N
R00439			✓	N
R00440			✓	N
R00441			✓	N
R00442			✓	N
R00443			✓	N
R00914		✓		P, S
R01119		✓		P

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

Reaction Code	multi-step	unclear reaction	no hint	elements appearing
R01122			✓	N
R01438	✓			N
R01725	✓			N
R01734		✓		O
R02127		✓		O, R
R02620			✓	R
R02713		✓		N
R02721		✓		N, P, S
R02811			✓	H
R02818	✓			S
R02822			✓	R
R02854		✓		P
R02904		✓		O, R
R02959		✓		R
R03141			✓	N, P, S
R03472	✓	✓		P
R03602		✓		O
R03603		✓		H, O
R03740	✓	✓		O
R03798	✓	✓		O
R03836		✓		O, R
R04808		✓		S
R04878		✓		S
R05044		✓		O, R
R05045		✓		O, R
R05188	✓			R
R05345		✓		O
R05473	✓			O
R05539			✓	H
R05666			✓	O
R05675			✓	P
R05721			✓	H
R05925			✓	R
R02334 / R06082			✓	R
R06133			✓	C

8.4 Transmutations

Reaction Code	multi-step	unclear reaction	no hint	elements appearing
R06195			✓	N, O
R06196	✓	✓		N, O
R06314		✓		O, P
R06315		✓		O, P
R06317		✓		O
R06320		✓		N
R06321		✓		O, P
R06352		✓		O
R06353		✓		O
R06397		✓		N, P, S
R06421			✓	P
R06443			✓	N, P
R06587			✓	P
R06672			✓	N
R06698			✓	N
R06724			✓	O
R06732	✓	✓		O
R06741			✓	O
R06744			✓	N
R06745			✓	N
R06746			✓	R, S
R06751			✓	R, S
R06756		✓		R, S
R06761		✓		N, R, S
R06765			✓	R, S
R06766			✓	Cl
R06768			✓	R, S
R06775			✓	R, S
R06901		✓		O
R06964	✓			O
R07007		✓		N
R07030		✓		O
R07097			✓	Cl
R07114	✓			N
R07115	✓			N

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

Reaction Code	multi-step	unclear reaction	no hint	elements appearing
R07545			✓	R
R07547			✓	R
R07548	✓	✓		O
R07552	✓	✓		O
R07555	✓	✓		O
R07838		✓		O, S
R07844	✓	✓		O
R07846	✓	✓		O
R07852	✓			O
R08027		✓		O
R08030		✓		O
R08035			✓	O
R08040		✓		N, O
R08078			✓	R
R08102			✓	R
R08122			✓	F
R08284		✓		O, P
R08299		✓		O
R08301		✓		O
R08302		✓		O
R08339	✓	✓		O
R08371	✓	✓		O
R08393	✓	✓		P
R08394	✓	✓		P
R08397	✓	✓		P
R08398	✓	✓		P
R08399	✓	✓		P
R08401	✓	✓		O
R08402	✓	✓		P
R08411	✓	✓		N
R08418	✓	✓		O
R08420			✓	O
R08421	✓	✓		O
R08427	✓	✓		O
R08428	✓	✓		O

8.4 Transmutations

Reaction Code	multi-step	unclear reaction	no hint	elements appearing
R08429	✓			O
R08431	✓	✓		O
R08435	✓	✓		N
R08480		✓		P
R08481	✓	✓		O
R08484	✓	✓		O
R08501	✓	✓		N, O
R08502	✓	✓		N, O
R08506	✓	✓		N
R08507	✓	✓		N
R08508	✓	✓		N
R08509	✓	✓		N, O
R08519		✓		P
R08524		✓		R
R08525		✓		R
R08535	✓	✓		O
R08579		✓		S
R08586			✓	N
R08609		✓		O
R08877			✓	N
R08882	✓	✓		N
R08883	✓			N
R08884	✓			N
R08885	✓			N
R08886	✓			N
R08988	✓	✓		O
R09051			✓	O
R09062	✓			R
R09089			✓	N
R09090	✓			N, P, S
R09094			✓	H
R09108		✓		O
R09112		✓		O
R09139		✓		Cl, H
R09142			✓	Cl, H

8. SUPPLEMENT: LIST OF PROBLEMATIC KEGG ENTRIES

Reaction Code	multi-step	unclear reaction	no hint	elements appearing
R09144			✓	Cl
R09145			✓	Cl
R09146			✓	O
R09147			✓	O
R09158		✓		Cl, H
R09181			✓	S
R09196			✓	R, S
R09305			✓	S
R09306			✓	R
R09309			✓	S
R09311			✓	R, S
R09315			✓	R
R09325			✓	R, S
R09326			✓	R, S
R09336			✓	N
R09340			✓	R, S
R09350			✓	R, S
R09449			✓	R
R09450			✓	R
R09467			✓	C, O
R09468			✓	H
R09470			✓	R
R09847		✓		R
R09851			✓	Cl
R10104			✓	R

9

SUPPLEMENT: SOURCE CODES FOR INTERVAL CHEMISTRY ALGORITHMS

9.1 Logarithmic Approximation

The following python source codes can be run within the IP[y] Notebook¹.

9.1.1 Initialization

```
%reset -s

pylab.rcParams['figure.figsize'] = (15.0, 10.0)
x = 5 # important natural constant

maximum=max
minimum=min
max=1
min=0

maxx=0
maxy=0
high=1
low=0
# Am_plus_n => Am + An
Am = [10,30]
An = [15,40]
Am_plus_n = [65,80] # Inflow
```

¹<http://ipython.org/notebook.html>

9. SUPPLEMENT: SOURCE CODES FOR INTERVAL CHEMISTRY ALGORITHMS

9.1.2 Calculation Methods

```
def processable():
    return [Am[min]+An[min], Am[max]+An[max]]

def unprocessed(inflow):
    prc=processable()
    result=[]
    if inflow[min]<prc[min]:
        result.append([inflow[min], prc[min]-1])
    if inflow[max]>prc[max]:
        result.append([prc[max]+1, inflow[max]])
    return result

def processed(inflow):
    prc=processable()
    if inflow[min]>prc[min]:
        prc[min]=inflow[min]
    if inflow[max]<prc[max]:
        prc[max]=inflow[max]
    if prc[min]>prc[max]:
        return []
    return prc

def range(lst):
    return lst[max]-lst[min]

def mergeoverlaps(a,b):
    if a[max]<b[min] or b[max]<a[min]:
        return [a,b]
    if a[min]<b[min]<=a[max]:
        if a[max]<b[max]:
            return [[a[min], b[max]]]
        else:
            return [a]
    if a[max]<b[max]:
        return [[b]]
    return [[b[min], a[max]]]

def outflow(inflow):
    prc=processed(inflow)
    if not prc:
        return []

    result_Am=Am[:]
    result_An=An[:]
    dn=Am[max]-Am[min]
    dn=An[max]-An[min]
    seed=Am[min]
    counter=0

    while Am[max]+result_An[min]<prc[min]:
        #print "increasing min An:", result_An[min], ">"
        dn=(1+dn)/2
        result_An[min]=result_An[min]+dn
        #print result_An[min]

        counter=counter+1
        if counter>20:
            break
    while Am[max]+result_An[min]>prc[min]:
        #print "too large, decreasing min An:", result_An[min], ">"
        dn=(1+dn)/2
        result_An[min]=result_An[min]-dn
        #print result_An[min]
        counter=counter+1
        if counter>20:
            break
```

9.1 Logarithmic Approximation

```
# at this point we should have a min result An

while An[max]+result_Am [min]<prc [min]:
    #print "increasing min Am:",result_Am [min],"=>"
    dm=(1+dm)/2
    result_Am [min]=result_Am [min]+dm
    #print result_An [min]

    counter=counter+1
    if counter >20:
        break
    while An[max]+result_Am [min]>prc [min]:
        #print "too large, decreasing min An:",result_An [min],"=>"
        dm=(1+dm)/2
        result_Am [min]=result_Am [min]-dm
        #print result_An [min]
        counter=counter+1
        if counter >20:
            break

# at this point we should have a min result Am

dm=Am[max]-Am [min]
dn=An [max]-An [min]

while Am [min]+result_An [max]>prc [max]:
    #print "decreasing max An:",result_An [max],"=>"
    dn=(1+dn)/2
    result_An [max]=result_An [max]-dn
    #print result_An [max]
    counter=counter+1
    if counter >20:
        break

    while Am [min]+result_An [max]<prc [max]:
        #print "too small, increasing max An:",result_An [max],"=>"
        dn=(1+dn)/2
        result_An [max]=result_An [max]+dn
        #print result_An [max]
        counter=counter+1
        if counter >20:
            break

# at this point we should have a max An

while An [min]+result_Am [max]>prc [max]:
    #print "decreasing max Am:",result_Am [max],"=>"
    dm=(1+dm)/2
    result_Am [max]=result_Am [max]-dm
    #print result_Am [max]
    counter=counter+1
    if counter >20:
        break

    while An [min]+result_Am [max]<prc [max]:
        #print "too small, increasing max Am:",result_Am [max],"=>"
        dm=(1+dm)/2
        result_Am [max]=result_Am [max]+dm
        #print result_Am [max]
        counter=counter+1
        if counter >20:
            break

# at this point we should have a max Am

#print result_Am
print [result_Am ,result_An]
return mergeoverlaps (result_Am ,result_An)
```

9. SUPPLEMENT: SOURCE CODES FOR INTERVAL CHEMISTRY ALGORITHMS

9.1.3 Main Entry Point

```
print "Am_plus_n = ", Am_plus_n
print "inflow = ", Am_plus_n
print "Am = ", Am
print "An = ", An
print "processable: ", processable()
print "processed:", processed(Am_plus_n)
print "unprocessed: ", unprocessed(Am_plus_n)
outflow=outflow(Am_plus_n)
print "outflow:", outflow
```

9.2 Geometrical Analysis

9.2.1 Initialization

```
%reset -s

pylab.rcParams['figure.figsize'] = (20.0, 10.0)
x = 5 # important natural constant

maximum=max
minimum=min
max=1
min=0

maxx=0
maxy=0
high=1
low=0
# Am_plus_n => Am + An
Pm = [10,30]
Pn = [15,40]
Px = [65,80] # Inflow
```

9.2.2 Main Code

```
import time
inflows = [Px]
ax = pyplot.figure().add_subplot(111)

pyplot.title('Intervals\n\n')
xmax=maximum([Px[max], Pn[max]+Pm[max]])+1
ymax=Pm[max]+1
feasible=[Pm[min]+Pn[min], Pm[max]+Pn[max]]
pyplot.xlim(0, xmax)
pyplot.ylim(0, ymax)

inflow=inflows[min]

# Am+n
plot([0, feasible[min]], [feasible[min], 0], 'k')
plot([0, feasible[max]], [feasible[max], 0], 'k')
text(feasible[min], -1, 'min valid P_k', color='k', horizontalalignment='center');
text(feasible[max], -1, 'max valid P_k', color='k', horizontalalignment='center');

# Am
plot([0, 1000], [Pm[max]]*2, 'b--')
plot([0, 1000], [Pm[min]]*2, 'b--')
text(-2.5, Pm[min], 'min P_m', color='b', horizontalalignment='right')
text(-2.5, Pm[max], 'max P_m', color='b', horizontalalignment='right')

# An
plot([Pn[min]]*2, [0, 1000], color='r', ls='--');
plot([Pn[max]]*2, [0, 1000], color='r', ls='--');
text(Pn[min]-1, ymax-0.5, 'min P_n', color='r', horizontalalignment='right');
text(Pn[max]-1, ymax-0.5, 'max P_n', color='r', horizontalalignment='right');
```

9.2 Geometrical Analysis

```

print_description=True
greens_recs=[]

def add_corner(x,y,num):
    corners.append([x,y])
    #text(x+0.3,y+0.3,str(num))

# plot inflow:
plot([0,inflow[min]],[inflow[min],0],'m--')
plot([0,inflow[max]],[inflow[max],0],'m--')
text(inflow[min]+1,-2,"inflow min",color="m",horizontalalignment='center')
text(inflow[max]+1,-2,"inflow max",color="m",horizontalalignment='center')
if inflow[max]>feasible[max]:
    ax.add_patch(Polygon([ [0,inflow[max]],[inflow[max],0],[feasible[max],0],[0,feasible[max]] ],color='r',
                        alpha=0.1))

if inflow[min]<feasible[min]:
    ax.add_patch(Polygon([ [0,inflow[min]],[inflow[min],0],[feasible[min],0],[0,feasible[min]] ],color='r',
                        alpha=0.1))

while inflows:
    inflow=inflows.pop()

    corners=[]

    if inflow[min]-Pm[max]>Pn[min]:
        add_corner(inflow[min]-Pm[max],Pm[max],1)
    else:
        if inflow[max]-Pm[max]>Pn[min]:
            add_corner(Pn[min],Pm[max],2)
        if inflow[min]-Pm[min]>=Pn[min]:
            add_corner(Pn[min],inflow[min]-Pn[min],3)

    if inflow[min]-Pn[max]>Pm[min]:
        add_corner(Pn[max],inflow[min]-Pn[max],4)
    else:
        if inflow[min]-Pm[min]>=Pn[min]:
            add_corner(inflow[min]-Pm[min],Pm[min],5)
        else:
            add_corner(Pn[min],Pm[min],6)

        if inflow[max]-Pm[min]>Pn[max]:
            add_corner(Pn[max],Pm[min],7)

    if inflow[max]>=Pm[max]+Pn[max]:
        add_corner(Pn[max],Pm[max],8)
    else:
        if inflow[max]-Pn[max]>=Pm[min]:
            add_corner(Pn[max],inflow[max]-Pn[max],9)
        else:
            add_corner(inflow[max]-Pm[min],Pm[min],10)

        if inflow[max]-Pm[max]>=Pn[min]:
            add_corner(inflow[max]-Pm[max],Pm[max],11)
        else:
            add_corner(Pn[min],inflow[max]-Pn[min],12)

    greens_recs.append(corners)

    maxm=Pm[min]
    minm=Pm[max]
    minn=Pn[max]
    maxn=Pn[min]
    for corner in corners:
        if corner[0]<minn:
            minn=corner[0]
        if corner[0]>maxn:
            maxn=corner[0]
        if corner[1]<minm:
            minm=corner[1]

```

9. SUPPLEMENT: SOURCE CODES FOR INTERVAL CHEMISTRY ALGORITHMS

```
    if corner[1]>maxm:
        maxm=corner[1]

ax.add_patch(Polygon([[0,minm],[maxn,minm],[minn,maxm],[0,maxm]],color='b',alpha=0.2))
# horizontal projection / m
ax.add_patch(Polygon([[minn,0],[maxn,0],[maxn,minm],[minn,maxm]],color='b',alpha=0.2))
# vertical projection / n

if minm<feasible[min]:
    if maxm<feasible[min]:
        ax.add_patch(Polygon([[0,minm],[minm,0],[maxm,0],[0,maxm]],color='r',alpha=0.1)) # diagonal/out of range
        continue
    else:
        ax.add_patch(Polygon([[0,minm],[minm,0],[feasible[min],0],[0,feasible[min]]],color='r',alpha=0.1))
        # diagonal/part out of range
        ax.add_patch(Polygon([[0,feasible[min]],[feasible[min],0],[maxm,0],[0,maxm]],color='#ffff00',alpha=0.4))
        # diagonal/part in range
else:
    ax.add_patch(Polygon([[0,minm],[minm,0],[maxn,0],[0,maxm]],color='#ffff00',alpha=0.4)) # diagonal/in range

if minn<feasible[min]:
    if maxn<feasible[min]:
        ax.add_patch(Polygon([[0,minn],[minn,0],[maxn,0],[0,maxn]],color='r',alpha=0.1)) # diagonal/out of range
        continue
    else:
        ax.add_patch(Polygon([[0,minn],[minn,0],[feasible[min],0],[0,feasible[min]]],color='r',alpha=0.1))
        # diagonal/part out of range
        ax.add_patch(Polygon([[0,feasible[min]],[feasible[min],0],[maxn,0],[0,maxn]],color='#ffff00',alpha=0.4))
        # diagonal/part in range
else:
    ax.add_patch(Polygon([[0,minn],[minn,0],[maxn,0],[0,maxn]],color='#ffff00',alpha=0.4))
    # diagonal/range

inflows.append([minm,maxm])
inflows.append([minn,maxn])

if print_description: #todo{description}
    for corners in greens_recs:
        ax.add_patch(Polygon(corners,color='#66ff66'))
    text((maxn+minn)/2+1,(maxm+minm)/2+1,'firing reactions',color='#005500',horizontalalignment='center');

show()
```

9.3 Fast Interval Expansion

9.3.1 Initialization

```
%reset -s

pylab.rcParams['figure.figsize'] = (15.0, 10.0)
x = 5 # important natural constant

maximum=max
minimum=min
max=1
min=0

maxx=0
maxy=0
high=1
low=0
# Am_plus_n => Am + An
Am = [4,12]
An = [20,30]
Am_plus_n = [25,33] # Inflow

inflows=[Am_plus_n]
```


9.3.2 Calculation Methods

```

def processable(): # constant runtime
    return [Am[min]+An[min], Am[max]+An[max]]

def unprocessed(inflow): # constant runtime
    prc=processable()
    result=[]
    if inflow[min]<prc[min]:
        result.append([inflow[min], prc[min]-1])
    if inflow[max]>prc[max]:
        result.append([prc[max]+1, inflow[max]])
    if (prc[max]<inflow[min]) or (inflow[max]<prc[min]):
        result=[inflow[:]]
    return result

def processed(inflow): # constant runtime
    prc=processable()
    if inflow[min]>prc[min]:
        prc[min]=inflow[min]
    if inflow[max]<prc[max]:
        prc[max]=inflow[max]
    if prc[min]>prc[max]:
        return []
    return prc

def mergeoverlaps(a,b): # constant runtime
    if a[max]<b[min] or b[max]<a[min]:
        return [a,b]
    if a[min]<b[min]<=a[max]:
        if a[max]<b[max]:
            return [[a[min], b[max]]]
        else:
            return [a]
    if a[max]<b[max]:
        return [[b]]
    return [[b[min], a[max]]]

def products(inflow): # constant runtime
    prc=processed(inflow)
    if not prc:
        return []
    Am.r=Am[:]
    An.r=An[:]
    if Am[min]+An[max]<=prc[max]:
        if Am[min]+An[max]<=prc[min]:
            Am.r[min]=prc[min]-An[max]
    else:
        An.r[max]=prc[max]-Am[min]

    if Am[max]+An[min]<=prc[max]:
        if Am[max]+An[min]<=prc[min]:
            An.r[min]=prc[min]-Am[max]
    else:
        Am.r[max]=prc[max]-An[min]
    return mergeoverlaps(Am.r, An.r)

def mergeIntervals(listA, listB): # O(len(listA)+len(listB))
    # clone, so the original lists are not changed
    listA=listA[:]
    listB=listB[:]
    result=[]
    last=None

    if not listA:
        return listB
    if not listB:
        return listA
    while listA or listB:
        if listA:

```

9. SUPPLEMENT: SOURCE CODES FOR INTERVAL CHEMISTRY ALGORITHMS

```
    if listB:
        a=listA [0]
        b=listB [0]
        if a [min]<b [min]:
            nxt=listA .pop (0)
        else:
            nxt=listB .pop (0)
    else:
        nxt=listA .pop (0)
else:
    nxt=listB .pop (0)

nxt=nxt [:] #clone!

if not result:
    result =[nxt]
else:
    last=result [-1]
    if nxt [min]<last [max]:
        last [max]=maximum (last [max] ,nxt [max])
    else:
        result .append (nxt)
return result
```

9.3.3 Main Entry Point

```
print "Am_plus_n = ",Am_plus_n
print "Am = ",Am
print "An = ", An
print ""
print "processable: ",processable ()
print

all_products=[]
dead_ends=[]

while inflows:
    inflow=inflows .pop (0)
    print "inflow = ",inflow
    print "processed:", processed (inflow)
    unprc=unprocessed (inflow)
    print "unprocessed: ",unprc
    prd=products (inflow)
    print "produces:", prd
    inflows=inflows+prd
    dead_ends=mergeIntervals (dead_ends ,unprc)
    all_products=mergeIntervals (all_products ,prd)
    print

print "all products:", all_products
print "dead ends:", dead_ends
```

10

CURRICULUM VITAE

10. CURRICULUM VITAE

Personal Data

Name Stephan Richter
Address Sophienstraße 10, 07743 Jena
Date of Birth 1983 October 26, 07381 Pößneck, Germany
Email s.richter@srssoftware.de

Education

1994 - 2002 Orlatalgymnasium, 07806 Neustadt an der Orla, Germany, with grade Abitur
2002 October - 2003 June National services in 92536 Pfreimd, Germany
2003 - 2010 Study of Bioinformatics at the Friedrich-Schiller-University Jena, Germany
2007 - 2009 Member of the student representative council
2010 April completed with diploma thesis “Discovery of biodegradation pathways by in-silico network evolution” with mark 1.5
2010 - 2014 PhD studies in the Bio Systems Analysis Research group of PD Dr. Peter Dittrich at the FSU, Jena with scholarship from the Helmholtz-Foundation. Cooperation with the Helmholtz Centre for Environmental Research in Leipzig, Germany.
2010 October Foundation of the SRSsoftware GbR company for side projects, Development of a biogas data analysis software (“Software zur Erfassung und Verwaltung der Daten experimenteller Biogasreaktoren”) for the company UGN Umwelttechnik GmbH, Gera

Research stays, Conferences, and Workshops

2011 March Poster presentation at the Keystone Symposium
“Microbial Communities as Drivers of Ecosystem Complexity”
in Breckenridge, Colorado, USA
2011 November Poster presentation at Heraeus Seminar
“Biothermodynamics of Metabolic and Ecological Networks”
2012 September Poster presentation at the
Cobra Summer School on Biological and Chemical IT
2013 March/April Research stay in the Laboratory of Chemical Life Science of
Professor Susumu Goto at the Institute for Chemical Research,
Kyoto University, Japan
2013 April Talk at the DocConference of the Helmholtz Centre for Environmental
Research, UFZ, Leipzig, Germany
2013 September Poster presentation at the “Metabolic Pathway Analysis”
Conference in Oxford, UK

REFERENCES

- [1] Alcántara, R., Axelsen, K. B., Morgat, A., Belda, E., Coudert, E., Bridge, A., Cao, H., de Matos, P., Ennis, M., Turner, S., Owen, G., Bougueleret, L., Xenarios, I. and Steinbeck, C. (2012), ‘Rhea—a manually curated resource of biochemical reactions.’, *Nucleic Acids Res* **40**(Database issue), D754–D760.
URL: <http://dx.doi.org/10.1093/nar/gkr1126> 56, 69
- [2] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M. and Sherlock, G. (2000), ‘Gene ontology: tool for the unification of biology. the gene ontology consortium.’, *Nat Genet* **25**(1), 25–29.
URL: <http://dx.doi.org/10.1038/75556> 56
- [3] Bartholomé, K., Rius, M., Letschert, K., Keller, D., Timmer, J. and Keppler, D. (2007), ‘Data-based mathematical modeling of vectorial transport across double-transfected polarized cells.’, *Drug Metab Dispos* **35**(9), 1476–1481. 140
- [4] Borenstein, E. and Feldman, M. W. (2009), ‘Topological signatures of species interactions in metabolic networks’, *Journal of Computational Biology* **16**(2), 191–200.
URL: <http://dx.doi.org/10.1089/cmb.2008.06TT> 28, 52
- [5] Borenstein, E., Kupiec, M., Feldman, M. W. and Ruppin, E. (2008), ‘Large-scale reconstruction and phylogenetic analysis of metabolic environments’, *Proceedings of the National Academy of Sciences* **105**(38), 14482–14487.
URL: <http://dx.doi.org/10.1073/pnas.0806162105> 28, 45, 52
- [6] Borghans, J. A., Dupont, G. and Goldbeter, A. (1997), ‘Complex intracellular calcium oscillations. A theoretical exploration of possible mechanisms.’, *Biophys Chem* **66**(1), 25–41. 19
- [7] Brock, W. H. (1997), *Viewegs Geschichte der Chemie*, B Kleidt, H Voelker.
URL: <http://www.springer.com/chemistry/book/978-3-540-67033-9> 52
- [8] Bungay, S. D., Gentry, P. A. and Gentry, R. D. (2003), ‘A mathematical model of lipid-mediated thrombin generation.’, *Math Med Biol* **20**(1), 105–129. 140
- [9] Bäck, T. (1996), *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*, Oxford University Press Oxford, UK. 47
- [10] Centler, F. and Dittrich, P. (2007), ‘Chemical organizations in atmospheric photochemistries - a new method to analyze chemical reaction networks’, *Planet Space Sci* **55**, 413–428. 6
- [11] Centler, F., Kaleta, C., di Fenizio, P. S. and Dittrich, P. (2008), ‘Computing chemical organizations in biological networks’, *Bioinformatics* **24**(14), 1611–1618. 6, 7, 15, 23, 131, 138

REFERENCES

- [12] Cottret, L., Vieira, M. P., Vicente, A., Alberto, M.-S., Leen, S., Hubert, C. and Marie-France, S. (2010), ‘Graph-based analysis of the metabolic exchanges between two co-resident intracellular symbionts, *Baumannia cicadellinicola* and *Sulcia muelleri*, with their insect host, *Homalodisca coagulata*’, *PLoS Comput Biol* **6**(9), e1000904.
URL: <http://dx.doi.org/10.1371/journal.pcbi.1000904> 52
- [13] Cottret, L., Vieira, M. P., Vicente, A., Alberto, M.-S., Viduani, M. F., Marie-France, S. and Leen, S. (2008), ‘Enumerating precursor sets of target metabolites in a metabolic network’, *Algorithms in Bioinformatics* **5251**, 233–244.
URL: <http://www.citeulike.org/user/pablocarb/article/8015886> 52
- [14] Covert, M. W., Schilling, C. H., Famili, I., Edwards, J. S., Goryanin, I. I., Selkov, E. and Palsson, B. Ø. (2001), ‘Metabolic modeling of microbial strains in silico’, *Trends in Biochemical Sciences* **26**(3), 179–186.
URL: <http://www.sciencedirect.com/science/article/pii/S0968000400017540> 52
- [15] de Figueiredo, L. F., Podhorski, A., Rubio, A., Kaleta, C., Beasley, J. E., Schuster, S. and Planes, F. J. (2009), ‘Computing the shortest elementary flux modes in genome-scale metabolic networks.’, *Bioinformatics* **25**(23), 3158–3165.
URL: <http://dx.doi.org/10.1093/bioinformatics/btp564> 45
- [16] de Lorenzo, V. (2008), ‘Systems biology approaches to bioremediation’, *Current Opinion in Biotechnology* **19**(6), 579–589.
URL: <http://dx.doi.org/10.1016/j.copbio.2008.10.004> 26, 52
- [17] Dittrich, P. and Speroni di Fenizio, P. (2007), ‘Chemical organization theory’, *Bull Math Biol* **69**(4), 1199–1231.
URL: <http://dx.doi.org/10.1007/s11538-006-9130-8> xv, 6, 7, 124
- [18] Drefahl, A. (2011), ‘Curlysmiles: a chemical language to customize and annotate encodings of molecular and nanodevice structures.’, *J Cheminform* **3**(1), 1.
URL: <http://dx.doi.org/10.1186/1758-2946-3-1> 94
- [19] Faeder, J. R., Blinov, M. L. and Hlavacek, W. S. (2009), ‘Rule-based modeling of biochemical systems with bionetgen.’, *Methods Mol Biol* **500**, 113–167.
URL: http://dx.doi.org/10.1007/978-1-59745-525-1_5 xv, 94, 120
- [20] Feinberg, M. and Horn, F. J. M. (1974), ‘Dynamics of open chemical systems and the algebraic structure of the underlying reaction network’, *Chem. Eng. Sci.* **29**(3), 775–787. 6, 124
- [21] Feist, A. M. and Palsson, B. Ø. (2010), ‘The biomass objective function.’, *Curr Opin Microbiol* **13**(3), 344–349.
URL: <http://dx.doi.org/10.1016/j.mib.2010.03.003> 46
- [22] Félix, L. and Valiente, G. (2007), ‘Validation of metabolic pathway databases based on chemical substructure search.’, *Biomol Eng* **24**(3), 327–335.
URL: <http://dx.doi.org/10.1016/j.bioeng.2007.02.008> 52, 64
- [23] Fontana, W. and Buss, L. W. (1994), ‘The arrival of the fittest’: Toward a theory of biological organization’, *Bull. Math. Biol.* **56**, 1–64. 7
- [24] Goldbeter, A. (1991), ‘A minimal cascade model for the mitotic oscillator involving cyclin and CDC2 kinase.’, *Proc Natl Acad Sci U S A* **88**(20), 9107–9111. 19
- [25] Green, M. L. and Karp, P. D. (2005), ‘Genome annotation errors in pathway databases due to semantic ambiguity in partial ec numbers.’, *Nucleic Acids Res* **33**(13), 4035–4039.
URL: <http://dx.doi.org/10.1093/nar/gki711> 52

REFERENCES

- [26] Handorf, T., Ebenhöf, O. and Heinrich, R. (2005), ‘Expanding metabolic networks: Scopes of compounds, robustness, and evolution’, *Journal of Molecular Evolution* **61**, 498–512. 52
- [27] Harris, L. A., Hogg, J. S. and Faeder, J. R. (2009), ‘Compartmental rule-based modeling of biochemical systems’, *Simulation Conference (WSC), Proceedings of the 2009 Winter* pp. 908 – 919.
URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5429719> 94
- [28] Hastings, J., de Matos, P., Dekker, A., Ennis, M., Harsha, B., Kale, N., Muthukrishnan, V., Owen, G., Turner, S., Williams, M. and Steinbeck, C. (2013), ‘The chebi reference database and ontology for biologically relevant chemistry: enhancements for 2013.’, *Nucleic Acids Res* **41**(Database issue), D456–D463.
URL: <http://dx.doi.org/10.1093/nar/gks1146> 56
- [29] Heller, S., McNaught, A., Stein, S., Tchekhovskoi, D. and Pletnev, I. (2013), ‘Inchi - the worldwide chemical structure identifier standard.’, *J Cheminform* **5**(1), 7.
URL: <http://dx.doi.org/10.1186/1758-2946-5-7> 80, 94
- [30] Hooke, R. (1665), *Micrographia: Or Some Physiological Descriptions of Minute Bodies Made by Magnifying Glasses, with Observations and Inquiries Thereupon*, Dover phoenix editions, Dover Publications.
URL: <http://books.google.de/books?id=0DYXk.9XX38C> 1
- [31] Howorth, M. and Soddy, F. (1953), *Atomic Transmutation: the greatest discovery ever made. From memoirs of Prof. Frederick Soddy.[With portraits.]*, New World Publications. 63
- [32] Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Novère, N. L., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J. and Forum, S. B. M. L. (2003), ‘The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models.’, *Bioinformatics* **19**(4), 524–531. iii, xv, 3, 6
- [33] James, C. A., Vandermeersch, T. and Dalke, A. (2007), ‘OpenSMILES specification’, online doc on SourceForge.net.
URL: <http://www.opensmiles.org/> 94
- [34] Kaleta, C., Centler, F., di Fenizio, P. S. and Dittrich, P. (2008), ‘Phenotype prediction in regulated metabolic networks.’, *BMC Syst Biol* **2**(1), 37. 6
- [35] Kaleta, C., Centler, F. and Dittrich, P. (2006), ‘Analyzing Molecular Reaction Networks: From Pathways to Chemical Organizations’, *Mol Biotechnol* **34**(2), 117–124. 20
- [36] Kaleta, C., Richter, S. and Dittrich, P. (2009), ‘Using chemical organization theory for model checking’, *Bioinformatics* **25**, 1915–1922. 6
- [37] Kanehisa, M. (1997), ‘A database for post-genome analysis’, *Trends Genet.* **13**, 375–376. xv
- [38] Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., Kawashima, S., Katayama, T., Araki, M. and Hirakawa, M. (2006), ‘From genomics to chemical genomics: new developments in KEGG’, *Nucleic Acids Research* .
URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1347464/> xv
- [39] Kanehisa, M., Goto, S., Sato, Y., Furumichi, M. and Tanabe, M. (2012), ‘KEGG for integration and interpretation of large-scale molecular data sets’, *Nucleic Acids Research* **40**, 109–114.
URL: <http://www.ncbi.nlm.nih.gov/pubmed/22080510> 34, 53, 56

REFERENCES

- [40] Karp, P. D. and Caspi, R. (2011), ‘A survey of metabolic databases emphasizing the MetaCyc family.’, *Arch Toxicol* **85**(9), 1015–1033.
URL: <http://dx.doi.org/10.1007/s00204-011-0705-2> 52
- [41] Kim, D., Rath, O., Kolch, W. and Cho, K.-H. (2007), ‘A hidden oncogenic positive feedback loop caused by crosstalk between Wnt and ERK pathways.’, *Oncogene* **26**(31), 4571–4579. ix, 11, 12, 15, 19, 22
- [42] Kumar, A., Suthers, P. F. and Maranas, C. D. (2012), ‘MetRxn: a knowledgebase of metabolites and reactions spanning metabolic models and databases’, *BMC Bioinformatics* **13**:6.
URL: <http://www.biomedcentral.com/1471-2105/13/6> 52
- [43] Kumar, V. S., Dasika, M. S. and Maranas, C. D. (2007), ‘Optimization based automated curation of metabolic reconstructions.’, *BMC Bioinformatics* **8**, 212. 23
- [44] Le Novère, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., Snoep, J. L. and Hucka, M. (2006), ‘BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems.’, *Nucleic Acids Res* **34**(Database issue), D689–D691. iii, 3, 5, 7, 11, 15
- [45] Li, C., Donizelli, M., Rodriguez, N., Dharuri, H., Endler, L., Chelliah, V., Li, L., He, E., Henry, A., Stefan, M. I., Snoep, J. L., Hucka, M., Novère, N. L. and Laibe, C. (2010), ‘BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models.’, *BMC Systems Biology* **4**, 92. 34
- [46] Marwan, W. (2003), ‘Theory of time-resolved somatic complementation and its use to explore the sporulation control network in *Physarum polycephalum*.’, *Genetics* **164**(1), 105–115. 20, 21
- [47] Marwan, W., Wagler, A. and Weismantel, R. (2008), ‘A mathematical approach to solve the network reconstruction problem’, *Mathematical Methods of Operational Research* **67**.
URL: <http://www.springerlink.com/content/w1211571x1451423/> 52
- [48] Matsumaru, N., Lenser, T., Hinze, T. and Dittrich, P. (2007), Toward Organization-Oriented Chemical Programming: A Case Study with the Maximal Independent Set Problem, in F. Dressler and I. Carreras, eds, ‘Advances in Biologically Inspired Information Systems’, Vol. 69 of *Studies in Computational Intelligence*, pp. 147–163. 6
- [49] Moore, G. E. (1965), ‘Cramming more components onto integrated circuits’, *Electronics* **38**(8).
URL: <http://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf> 2
- [50] Olsen, L. F., Kummer, U., Kindzelskii, A. L. and Petty, H. R. (2003), ‘A model of the oscillatory metabolism of activated neutrophils.’, *Biophys J* **84**(1), 69–81. 17, 19
- [51] Orth, J. D., Thiele, I. and Palsson, B. Ø. (2010), ‘What is flux balance analysis?’, *Nature Biotechnology* **28**(3), 245–248. iii, xv, 3, 28, 52
- [52] Ott, M. A. and Vriend, G. (2006), ‘Correcting ligands, metabolites, and pathways.’, *BMC Bioinformatics* **7**, 517.
URL: <http://dx.doi.org/10.1186/1471-2105-7-517> 52, 61
- [53] Pazos, F., Valencia, A. and Lorenzo, V. D. (2003), ‘The organization of the microbial biodegradation network from a systems-biology perspective.’, *EMBO Rep* **4**(10), 994–999.
URL: <http://dx.doi.org/10.1038/sj.embor.embor933> 29
- [54] Reed, J. L., Patel, T. R., Chen, K. H., Joyce, A. R., Applebee, M. K., Herring, C. D., Bui, O. T., Knight, E. M., Fong, S. S. and Palsson, B. Ø. (2006), ‘Systems approach to refining genome annotation.’, *Proc Natl Acad Sci U S A* **103**(46), 17480–17484. 23

REFERENCES

- [55] Rehm, M., Huber, H. J., Dussmann, H. and Prehn, J. H. M. (2006), ‘Systems analysis of effector caspase activation and its control by X-linked inhibitor of apoptosis protein.’, *EMBO J* **25**(18), 4338–4349. 140
- [56] Rohwer, J. M. (2012), ‘Kinetic modelling of plant metabolic pathways.’, *J Exp Bot* **63**(6), 2275–2292.
URL: <http://dx.doi.org/10.1093/jxb/ers080> 52
- [57] Schellenberger, J., Park, J. O., Conrad, T. M. and Palsson, B. Ø. (2010), ‘Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions.’, *BMC Bioinformatics* **11**, 213.
URL: <http://dx.doi.org/10.1186/1471-2105-11-213> 34
- [58] Schilling, C. H., Letscher, D. and Palsson, B. Ø. (2000), ‘Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective.’, *J Theor Biol* **203**(3), 229–248. xv, 5, 19, 123
- [59] Schuster, S., Dandekar, T. and Fell, D. A. (1999), ‘Detection of elementary flux modes in biochemical networks: A promising tool for pathway analysis and metabolic engineering.’, *Trends Biotechnol* **17**(2), 53–60. 19, 123
- [60] Schuster, S., Fell, D. A. and Dandekar, T. (2000), ‘A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks.’, *Nat Biotechnol* **18**(3), 326–332.
URL: <http://dx.doi.org/10.1038/73786> 107, 110
- [61] Singh, A., Jayaraman, A. and Hahn, J. (2006), ‘Modeling regulatory mechanisms in IL-6 signal transduction in hepatocytes.’, *Biotechnol Bioeng* **95**(5), 850–862. 18, 19
- [62] Tomar, N. and De, R. K. (2013), ‘Comparing methods for metabolic network analysis and an application to metabolic engineering.’, *Gene* **521**(1), 1–14.
URL: <http://dx.doi.org/10.1016/j.gene.2013.03.017> 52
- [63] Trigo, A., Valencia, A. and Cases, I. (2009), ‘Systemic approaches to biodegradation.’, *FEMS Microbiol Rev* **33**(1), 98–108.
URL: <http://dx.doi.org/10.1111/j.1574-6976.2008.00143.x> 26
- [64] Trinh, C. T., Wlaschin, A. and Sreenc, F. (2009), ‘Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism.’, *Appl Microbiol Biotechnol* **81**(5), 813–826.
URL: <http://dx.doi.org/10.1007/s00253-008-1770-1> xv, 5
- [65] Varma, A. and Palsson, B. Ø. (1994), ‘Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use’, *Bio/Technology* **12**(10), 994–998. 19, 123
- [66] Weininger, D. (1988), ‘SMILES, a chemical language and information system’, *Journal of Chemical Information and Computer Sciences* **28**(1), 31–36.
URL: <http://pubs.acs.org/doi/abs/10.1021/ci00057a005> 80, 94
- [67] Wixon, J. and Kell, D. (2000), ‘The Kyoto encyclopedia of genes and genomes–KEGG.’, *Yeast* **17**(1), 48–55.
URL: <http://dx.doi.org/3.0.CO;2-H> 53
- [68] Yamada, S., Shiono, S., Joo, A. and Yoshimura, A. (2003), ‘Control mechanism of JAK/STAT signal transduction pathway.’, *FEBS Lett* **534**(1-3), 190–196. 17, 19

ehrenwörtliche Erklärung

Hiermit erkläre ich,

- dass mir die Promotionsordnung der Fakultät für Mathematik und Informatik bekannt ist,
- dass ich die vorliegende Dissertation selbst angefertigt habe und dabei keine Textabschnitte oder Ergebnisse eines Dritten oder von eigenen Prüfungsarbeiten ohne Kennzeichnung übernommen habe und alle von mir benutzten Hilfsmittel, persönlichen Mitteilungen und Quellen in meiner Arbeit angegeben habe,
- dass ich die Hilfe eines Promotionsberaters nicht in Anspruch genommen habe und dass Dritte weder mittelbar noch unmittelbar geldwerte Leistungen von mir für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der Dissertation stehen,
- dass ich die Dissertation nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe.

Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts haben mich folgende Personen unterstützt:

PD Dr. Peter Dittrich, Dr. Martin Thullner, Dr. Ingo Fetzer, Dr. Florian Centler sowie Dipl. Math. Peter Kreyssig

Ich habe weder die gleiche noch eine in wesentlichen Teilen ähnliche Abhandlung bei einer anderen Hochschule als Dissertation eingereicht.

Jena, 7. Juli 2015