A Formal Model of Molecular Codes with Respect to Chemical Reaction Networks

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
Diplom-Bioinformatiker Dennis Görlich
geboren am
02. Juni 1983 in Hagen

Gutachter
1. PD Dr. Peter Dittrich (Friedrich-Schiller-Universität Jena)
2. PD Dr. Stefan Artmann (Friedrich-Schiller-Universität Jena)
3. Prof. Dr. Marcello Barbieri (Università di Ferrara)

Tag der öffentlichen Verteidigung: 19.04.2013

Abstract

The present thesis introduces a theory of molecular codes with respect to chemical reaction networks. Codes, in general, are mappings between sets of entities. Encoding is very well known in many disciplines, like language, where concepts are said to be encoded in words or spoken language, and computer science where, e.g. commands have to be encoded into binary digits for execution, or optimal codes for data compressing have to be developed. In biology the notion of codes has been largely introduced together with the discovery of the gene translation mechanisms, i.e. the genetic code. Recent developments in molecular and cellular biology postulate other molecular codes beside the genetic code, e.g. the histone code or the sugar code. In the literature these codes are described in detail in their biochemical mechanisms, but the usage of the term "code" is ambiguous. Often "code" denotes only the codewords, e.g. combinations of covalent histone modifications, but neglects the mapping between codewords and their "meanings". It is also not yet clear which biological relevant entities (processes, molecular species, system states) are encoded by these novel codes. One reason for the unclear usage of the code concept is the lack of an objective definition of a "molecular code" applicable to biological systems. To enable molecular biology to properly analyse molecular codes a formal, objective and testable definition of code is necessary. In this thesis I will present a formal concept of molecular codes as mappings between sets of molecular species that are elements of a chemical reaction network, i.e. a model of a (bio-)chemical system.

An important property of a code is its contingency, i.e. the relations between codewords and their "meanings" could, in principle, be different. This should also hold for molecular codes to distinguish them from fixed mappings and to enable evolution to act on codes. Due to the contingency condition codes always occur as collection of (potential) mappings. These differ in their actual relations, but map the same sets of molecular species. The general definition of molecular codes as contingent molecular mappings is specialised by analysing binary molecular codes, i.e. codes between sets of only two molecular species. Furthermore, the definition of codes allows to analyse the properties of molecular codes, especially the relations between codes. I will analyse code nesting and code linkage as two forms of code relations. Both concept allow to describe cells as systems of codes.

Based on the definition of molecular codes it is possible to develop algorithms to identify codes in chemical reaction networks. I propose two different algorithms based on different structural network properties, i.e. on closed sets and paths, respectively. Both algorithms follow a brute force strategy and are computational not feasible for large networks. For the path algorithm I propose two heuristic variants, i.e. (1) using the k-shortest paths (instead of all paths), and (2) applying a Monte-Carlo-type subnetwork sampling with subsequent code analysis. The two heuristics do not guarantee to identify all codes, but generate an estimate on the number of codes. This approach is suited for large scale networks, as demonstrated for the metabolic network of cells and the human signal transduction network.

The algorithms are applied to a number of different reaction networks modelling combustion chemistries, a planetary photo chemistry, the gene translation system, the gene regulatory network, signalling by phosphorylation cascades, and two large scale biological networks obtained from databases. The analysis of these networks shows that abiotic networks do not have the ability to realize codes, while the biochemical systems do have the ability to implement molecular codes. The example of a phosphorylation cascade

network model shows the restriction to the structural approach of code identification, since here codes can only be implemented when the species' concentration is considered. Random networks are analysed as a null model of molecular codes. A statistical model is fitted that describes the number of molecular codes dependent on network size and network density. The analysis also shows that there exist an optimal interval for codes for a fixed network size. Very sparse networks and very dense networks do not allow for molecular coding. The optimal interval gives the network densities that allow for a large number of codes, assuming completely random processes of network generation. The analysis of an artificial chemistry shows that also a dense network can have codes. A randomisation study of this network results in a decrease in the number of codes, i.e. the network converges towards the null model. Similarly, we can assume that the number of codes could increase under random variation if the network is in the optimal interval.

From a theoretical point of view the ability to implement codes can be interpreted as semantic capacity. By identifying potential molecular codes a measure for the semantic capacity of (bio-)chemical systems is provided. Based on this notion hypotheses can be formulated with respect to the semantic capacity of biological systems, e.g. cells evolve towards higher semantic capacity, by employing subnetworks (subchemistries) that allow for coding. The results of this thesis will not answer this question completely, but give first results.

In the thesis I will also discuss how the static, semantic aspect of molecular codes can be (and has to be) supplemented by the pragmatic level, e.g. by including kinetics and probabilities. The inclusion of dynamics also allows to identify codes between whole system states.

Zusammenfassung

In der vorliegenden Dissertation führe ich ein formales Konzept für molekularer Kodes in chemischen Reaktionsnetzwerken ein. Kodes sind Abbildungen zwischen Mengen von Objekten. Kodierung ist ein verbreitetes Konzept. In der Linguistik wird der Zusammenhang zwischen Wörtern und den bezeichneten Objekten als Kodierung aufgefasst. In der Informatik werden Instruktionen in Bitstrings kodiert werden, bzw. optimale Kodes für Dateikomprimierung entwickelt. In der Biologie wurde das Kodekonzept zusammen mit der Entdeckung der Mechanismen der Gentranslation eingeführt, der genetische Kode. Die weitere Forschung in der Zell- und Molekularbiologie postuliert die Existenz weiterer Kodes in der Zelle neben dem genetischen Kode. Der Histone- und der Zuckerkode sind hier Beispiele. Diese neuartigen Kodes wurden bisher sehr detailiert in ihren biochemischen Mechanismen beschrieben, aber nutzen Unterschiedliche Definitionen des Kodebegriffs. Oft wird der Begriff "Kode" zur Bezeichnung der Kodewörter, zum Beispiel die Kombination verschiedener kovalenter Histonemodifikationen, verwendet, während die Bedeutung im Sinne einer Abbildung vernachlässigt wird. Dabei ist es auch nicht klar zwischen welchen Mengen (Prozesse, molekulare Spezies, Systemzustände) abgebildet wird. Ein Grund für die unklare Verwendung des Kodebegriffs ist das Fehlen einer objektiven Definition, die es erlaubt molekulare Kodes in biologischen Systemen zu erkennen. Eine formale, objektive und prüfbare Definition ist daher notwendig. Das Kodekonzept, das hier vorgestellt werden soll, basiert auf Modellen chemischer Systeme in Form von chemischen Reaktionsnetzwerken.

Ein wichtiger Aspekt von Kodes im allgemeinen ist Kontingenz. Eine kontingente Abbildung erlaubt es die Kodewörter und deren Bedeutungen willkürlich zuzuordnen, d.h. eine beobachtete Abbildung könnte prinzipiell auch in anderer Ausprägung vorliegen. Dies soll auch für molekulare Kodes gelten. Molekulare Kodes unterscheiden sich dadurch von feste Abbildungen und können als Ziel eines evolutionären Selektionsdrucks fungieren. Die Kontingenzbedingung bewirkt, dass Kodes immer als Menge vieler (potentieller) Kodes auftreten. Diese Kodes unterscheiden sich in ihren Beziehungen, aber bilden zwischen den selben Mengen ab. Ein Spezialfall der allgemeinen Definition molekularer Kodes stellt die Analyse binärer molekularer Kodes dar. Dies sind molekulare Kodes, die zwischen binären Mengen abbilden. Die Definition molekularer Kodes erlaubt außerdem die Analyse bestimmter Kodeeigenschaften, zum Beispiel Relationen zwischen Kodes. Ich habe in diesem Zusammenhang verschachtelte Kodes (code nesting) und zwei Formen der Kodeverknüpfung (code linkage) untersucht. Die Verwendung dieser Eigenschaften ermöglicht es die Zelle als System molekularer Kodes zu beschreiben.

Basierend auf der Definition ist es möglich Algorithmen zur Kodeidentifikation in chemischen Reaktionsnetzwerken anzugeben. Ich stelle zwei Algorithmen vor, die unterschiedliche Netzwerkeigenschaften ausnutzen, zum Einen geschlossene Mengen und zum Anderen die Pfade durch das Netzwerk. Beide Algorithmen folgen einer brute-force Strategie und sind für große Netzwerke sehr rechenintensiv. Für den Pfadalgorithmus stelle ich zwei Heuristiken vor. Die erste Heuristik verwendet die K kürzesten Pfade, während die zweite Heuristik zusätzlich in einem Monte-Carlo Ansatz Teilnetzwerke ermittelt, die anschließend mit dem Kodealgorithmus analysiert werden. Die entwickelten Algorithmen werden auf verschiedene Netzwerkmodelle angewandt: Verbrennungschemien, eine planetare Photochemie, das Gentranslationssystem, genregulatorische Netzwerke, Signalweiterleitung durch Phosporylierungskaskaden und zwei große biologische Netzwerke (Metabolism und Signaltransduktion) die aus Netzwerkdatenbanken stammen. Die

Analyse dieser Netzwerke zeigt dass abiotische Netze keine Kodes besitzen, während die biologischen Netzwerkmodelle sehr viele molekulare Kodes implementieren können. Das Beispiel der Phosphorilierungkaskaden zeigt aber auch die Grenzen dieses Ansatzes, da hier Konzentrationen zur Kodeidentifizierung hinzugezogen werden müssen. Zufällige Reaktionsnetzwerke können als Nullmodell für molekularer Kodes dienen, indem ein statistisches Modell angelernt wird, das die Anzahl molekularer Kodes in Abhängigkeit der Netzwerkgröße und Dichte beschreibt. Die Analyse der Daten zeigt auch, dass es ein optimales Interval (bezogen auf die Netzwerkdichte) für molekulare Kodes gibt. Sehr dünne und sehr dichte Netzwerke erlauben demnach keine Realisierung molekularer Kodes. Das optimale Interval gibt an welche Netzwerkdichten die Realisierung vieler molekularer Codes erlauben, unter der Anahme einer komplett zufälligen Netzwerkgenerierung. Die Analyse einer künstlichen Chemie zeigt, dass auch dichte Netzwerke Kodes enthalten können. Die Randomisierung dieses Netzwerks führt zu einer Verringerung der Kodierungskapazität, das Netztwerk konvergiert gegen das Nullmodell. Daran angelehnt kann die Hypothese aufgestellt werden, dass die Anzahl molekularer Kodes ansteigen kann, wenn das Netzwerk sich im optimalen Interval befindet. Die Fähigkeit eines Systems molekulare Kodes zu implementieren kann als semantische Kapazität aufgefasst werden, da ein Kode Zeichen und Bedeutungen miteinander verknüpft. Die Identifizierung molekularer Kodes liefert daher ein Maß für die semantische Kapazität eines Systems. Darauf basierend können Hypothesen in Bezug auf die semantische Kapazität biologischer Systeme formuliert werden, zum Beispiel, dass Zellen im Laufe ihrer Evolution mehr Subsysteme hoher semantischer Kapazität verwenden. Die vorliegende Arbeit wird diese Frage nicht abschließend beantworten, sondern liefert erste Resultate. Zum Ende der Arbeit diskutiere ich die Notwendigkeit den hier vorgestellten statischen Ansatz durch pragmatische Aspekte, d.h. Dynamik, Kinetiken und Wahrscheinlichkeiten, zu erweitern. Die Erweiterung um dynamische Aspekte ermöglicht zum Beispiel die Identifizierung von Kodes zwischen Systemzuständen.

Acknowledgements

First of all I want to thank Peter Dittrich for giving me the opportunity to do a PhD in his group and for finding time to discuss new ideas and to give support and advice. I also want to thank Stefan Artmann for all the discussions and input, especially, at the beginning of my project. Stefan Heinemann, as member of my JSMC thesis committee, for finding time for our meetings and for giving valuable input. My thanks goes to the members of the Bio Systems Analysis Group for providing an open ear for new ideas, for interesting discussions, for giving support and for almost always sharing their sweets. I want to thank Konstantin Riege who helped at the implementation of the random subnetwork sampling algorithm. I also want to thank Conny Müsse and Kathrin Schowtka for helping me through the university's bureaucracy. The support of the faculty's computer center staff was always appreciated to overcome minor and major IT issues.

I had the luck to be supported by a stipend of the excellence initiative graduate school "Jena School for Microbial Communication (JSMC)", which allowed many freedoms that would not be possible with other forms of funding. As JSMC fellow representative I want to thank the teams of representatives I had the luck to work in: The first team of representatives Nadine and Anne, the follow-up team Markus and Cris and the new team Sarahi, Markus and Martin, and Frank our long term JSMC representative. I also want to thank the organising teams of our conference "International Student Conference on Microbial Communication (MICOM)" which we started in 2010. Organising this conference was a lot of work (especially the first time), but also was lot of fun and yielded lots of experiences. Special thanks go to Carsten Thoms and Ulrike Schleier from the JSMC management. Both did and do an extraordinary job, and without their work JSMC would not be as successful and well organised as it is.

Finally, I want to thank my family for their ongoing support. My parents and parents-in-law for giving all kinds of support. My wonderful son Linus for being just as he is and with whom I will start many new adventures in future. My last and deepest thanks go to my wonderful wife Stephanie who always encourages me to go on and focus on the important things.

Contents

Chapter 1

Introduction

1.1 Biological information processing

Research of the last decades showed that cells communicate and process information [1]. This is not only true for human cells, where, for example, the hormone system is well known, but also for all other eukaryotic and prokaryotic species. While communication refers to an interaction between individual cells, information processing is a more general concept. The genetic system, implemented in every cell, maintains the blueprint for the cell's components, e.g. proteins. This stored information is utilised by the processes usually referred to as transcription and translation, in the case of proteins. Beside the genetic system cells maintain complex signal transduction networks that enables them to integrate information about their environment, internal state and incoming signals. This information is mainly used to regulate the cell's behaviour, i.e. to change the internal state.
The understanding of biological information processing is not only relevant as basic research, but can have direct practical applications, for example, to identify targets in the treatment of microbial infections [2]. From a theoretical point of view it is also of interest if different subsystems (biochemical systems) of cells are better suited to be used for information processing.

Syntax, semantics and pragmatics For theoretical analysis of biological information Shannon's theory of communication [3] has been applied successfully in various domains, like gene regulatory networks [4], bacterial quorum sensing [5], or signalling in molecular systems [6, 1]. The mathematical theory of communication focusses on uncertainty of events and intentionally neglects semantic aspects of information, because "*they are irrelevant for the engineering problem*" (Shannon [3], p. 1). In order to obtain a full understanding of biological information, studying semantic as well as pragmatic aspects would be important, if not necessary [7, 8].
The terms syntax, semantics and pragmatics [1] are concepts borrowed from the fields of language and semiotics. The transfer from these fields of study to the life sciences needs to be justified. Whether the linguistic terms used in biology are ill-posed or valuable concepts is discussed [10, 11]. These concepts have explanatory power in biological systems as discussed, for example, in [12]. The analogy between communication processes in language and semiotics, and molecular communication (where signals are

[1] For a detailed introduction to syntax, semantics, and pragmatics as semiotic concepts see for example [9].

mainly molecular species) is very strong: For example, in the case of microbial communication molecular species (signals), like acyl homoserine lactone (AHL) derivates, are constantly secreted into the environment by cells (sender). The receiving cells (receiver) maintain a receptor protein that regulates target genes in correlation with the signal's concentration. This communication behaviour is referred to as quorum sensing [13]. The sender (cell) encodes its internal state into a signalling molecule (AHL), sending it via a channel (diffusion in the environment), while the receiver (cell) decodes the signal by recognition at the receptor protein and triggering of subsequent events (change of internal state). This behaviour corresponds with the classical model of a communication process as presented by Shannon [3] (Figure 1.1). The syntactic level is given by the actual signalling molecule, or combinations thereof, i.e. the (encoded) message in Shannon's model. The semantic level is given by the encoding and decoding function and the pragmatic level describes when the communication is applied.

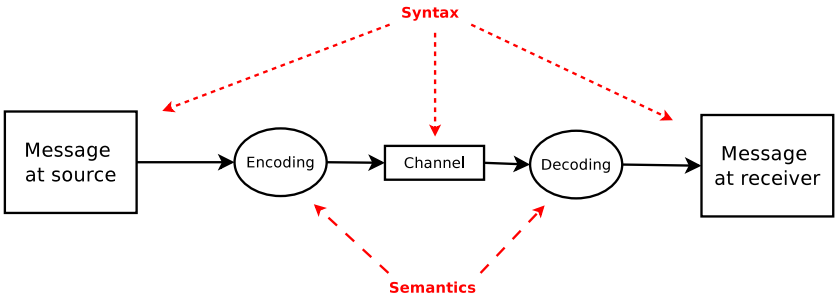


Figure 1.1 Shannon's communication model. A message is encoded by the sender, transmitted via a channel and decoded by the receiver (after [3]). The syntactic analysis mainly focuses on the (encoded) messages send via the channel. Semantics is related to the codes between sign and meaning. Encoding and decoding can be both analysed from a semantic perspective

In order to properly use semiotic concepts in biology we should provide a link to the realm of physics by (1) selecting an experimentally grounded and reliable formal description of the targeted biological system, by (2) providing precise, not necessarily formal, definitions of the semiotic concepts that shall be applied to the system, and by (3) interpreting these definitions by linking them to the formal description of the biological system.

While syntax refers to the internal organisation of a message, or signal [14], semantics refers to the relation between a sign and its meaning, i.e. a code [15, 16]. For example, the genetic code is a mapping between codons and amino acids [17], which is realised in cells by a complex translation machinery. An important property of a code is its contingency [18, 15], i.e. a type of inherent indeterminacy (cf. [18]). A relation between signs and meanings is said to be contingent, if it could be different. Different in the sense, that among the same sets of signs and meanings the individual elements could be related in a different way. This relation is not determined by the signs and meanings alone [7, 16]. In particular, this implies that natural laws allow to derive the relation only by knowing the context under which the signs are "interpreted". Furthermore, it implies the existence of another context under which the signs are "interpreted" differently. A code cannot be explained by physical laws [19], like the natural laws do not help in understanding the written law or the grammar of a language.

For biological systems, which are mainly governed by physical and chemical laws, con-

tingency (sometimes called arbitrariness) need not necessarily to hold, but it is discussed whether it is a useful concept [7, 20, 21, 18, 22]. While in language it comes naturally to us that we can change the object we denote by a word easily, in molecular systems we first have to understand the nature of the relation between signs and meanings. Contingency in molecular systems seems to stand in contrast to the rules of physics and chemistry which govern all molecular processes, because if the laws of physics explain every process there would be no place for a contingency. The example of the genetic code shows that this is not always the case. The relation between codons and amino acids is realised by a sequence of reactions that are governed by chemical rules, but the choice which codon is translated into which amino acid can be understood as arbitrary, or contingent. If we say codons (signs) are mapped to amino acids (meanings) then a (total) arbitrary mapping could in principle relate all signs to all meanings. This freedom of assignment is also a property of the chemical system. There may be constraints to the actual shape of the mapping, but as long as in principle the mapping could be changed it can be considered to be contingent. Assuming a total contingent relation between signs and meanings is the most general state we can describe in this context. Barbieri identified these as (chemical) "independent worlds" [16]. The contingency is implemented in the structure of the adapter molecules that allows to connect these two worlds.
In biological systems signs and meanings are molecular species (cp. [16, 23]). Contingency in a biological system needs to be identified among the relations between the molecular species in order to characterise a code, the semantic level of the biological system.

1.2 Related formal concepts

I will briefly review the concepts of code as used in Shannon's "Theory of communication", Tlusty's "molecular codes", and Barbieri's "organic codes".

The notion of code in information theory and coding theory. The first notion of code is often used when a combinatorial complexity is described, as for example the codons of the genetic code. This notion is related to the definition of "code" as used in coding theory, a discipline of discrete mathematics. Coding theory studies the construction, parametric bounds, and implementation of (error-correcting) codes. In coding theory a code $\mathcal{C}$ is a set of codewords from a common alphabet, $\mathcal{C} \subset A^*$ (cp. [24]). Certain other conditions can be applied to such a code, for example, fixed length code words, as for block codes. Implicitly, these codewords are situated in a communication process between a sender, who needs to encode a message that has to be sent via a channel, and a receiver who needs to decode it. While coding theory mainly focusses on the structure and properties of the codewords, the second notion of code (code = mapping) refers to the process of encoding (decoding). It catches the relation between a codeword and its "meaning".
Information theory utilises the second notion of code. Cover and Thomas, for example, defined a (source) code "[..] C for a random variable X [as] a mapping from [..] the range of X, to [..] the set of finite length strings of symbols from a D-ary alphabet." [25]. This definition describes the encoding and is used, for example, in data compression. Alternatively, the decoding scheme is a mapping from the codewords to the "message".

In Shannon's "Theory of communication" [3] the messages to be send through the channel are encoded before sending. The meaning of each message is irrelevant to the function of the channel, and thus is also not captured by Shannon's theory. The code, i.e. the mapping between message and the encoded string of binary digit, keeps some importance, e.g. it can be optimised with respect to the properties of the channel. Shannon's source coding theorem, for example, shows that the average number of bits per symbol (of the message) cannot be smaller than the channel's entropy [3]. In computer science and mathematics "coding theory" has been established as a field of study. It deals with the engineering problem to identify optimal codes for applications in data compression, cryptography, or error-correction (cf. [26, 27] and references therein).
Beside this, the notion of code has been applied to biological research to understand how information encoding in biological systems is employed.

A physical model molecular codes Tlusty describes molecular codes from a theoretical, physical point of view [28]. In his framework he defines the sets of signs and meanings beforehand and generally allows all signs to be mapped onto all meanings. This mapping is modelled as a transition matrix that gives the probabilities that a sign a is mapped onto a meaning ω. The process of encoding and decoding is modelled as a Markov chain (see Figure 1.2). By defining cost and quality of a code he was able to show that coding occurs as a phase transition[29]. The optimisation of the code via the transition matrix accesses the semantic level (mapping between signs and meanings) from the pragmatic level (optimality, fitness). The coding state can be reached from a random, non-coding state by either increase in gain (bits of information to increase code quality), an increased reading accuracy of the signals, a larger distance between the meanings, or increase of population size [29].

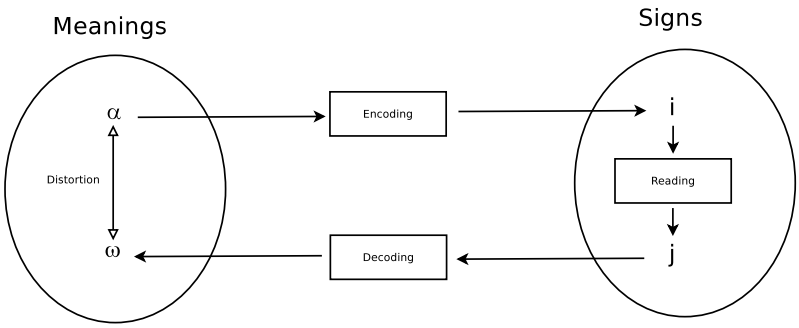


Figure 1.2 Molecular code framework by Tlusty. In Tlusty's framework of molecular codes a set of meanings can be encoded by a set of signs and be decoded. The whole process can be modelled as Markov process representing en- and decoding, as well as reading as transition matrices. Eventually, the distortion between two meanings can be used as a measure for the code's fitness. After [28].

Vestigian and colleagues [30] modelled the genetic code as probabilistic map, similarly to Tlusty's approach. In their formulation the probability that a codon c is mapped to an amino acid α is the sum over all probabilities that c is read by a tRNA t multiplied by the probability that t is charged with α. In their work ([30]) they showed that horizontal gene transfer may have played a major role in the evolution of the genetic code. This result also is situated on the pragmatic level (how does the code evolve).

Organic codes Barbieri introduced the concept of "organic codes" [31] as a semiotic framework to explain the sign usage in biological systems. His definition of code requires three propositions to be met: There have to exist (1) two independent molecular worlds that (2) are connected by a system of adapters that realise a (3) relation between elements of the two worlds [16]. Independent molecular worlds, here, are characterised by chemically different molecular species, as for example in the genetic code where DNA is chemically different from the amino acids. This also implies that there is no direct chemical relationship between these worlds, e.g. metabolic reactions. By his notion of "independent worlds" a relation between signs and meanings always needs to be contingent, because if the worlds are independent no chemical or physical law determines the mapping. The relation that is made between signs and meanings, i.e. the code, is realised by the adapters. To identify an organic code the adapter molecules have to be identified. An adapter molecule performs two independent recognition processes that link the two independent worlds. The genetic code, as organic code, connects DNA and amino acids (independent worlds), via the action of tRNAs. A tRNA molecule recognises the (complementary) RNA codon (first recognition) and carries the appropriate amino acids (second recognition). There exist a system of tRNAs that, taken together, implement the genetic code. The concept can be applied to other cellular subsystems, like splicing [31, 16].

The need for a formal definition of molecular codes Tlusty's framework of molecular codes allows to derive general properties with respect to a code's evolution and fitness. But is does not help to identify a chemical system that allows for coding. Barbieri's concept of organic codes, in principle, allows for the identification of a code when the independent world and the adapters can be identified. Nevertheless, a more formal definition of molecular codes, that objectively can identify potential codes in chemical system, would be the next important step towards a code-based analysis of biological systems.
In this thesis I will present a formal concept of molecular codes based on chemical reaction networks. Chemical reaction networks are discrete models of actual biological or chemical systems. The grounding of a formal definition of molecular codes in an explicit formal model of a system is, to the current state of the art, new.
With this approach, the semiotic concept of code gets – at least partially – operationalised by means of physical experiments. In particular, it allows to incorporate contingency in a formal model of molecular codes.

1.3 Structure of the thesis

In the present chapter I gave a general introduction to the background of biological information processing and the motivation to develop formal models of otherwise loose concepts. In Chapter 2 I will review three major biological systems that have been reported to constitute a molecular code, i.e. the genetic code, the histone code, and the sugar code. The chapter once again motivates the need for a more formal definition of codes. Especially, in the histone code and the sugar code the notion of code is not used homogeneously. In Chapter 3 I will present the definition of molecular codes with respect to chemical reaction networks. I will also describe algebraic properties of molecular codes. The formal definition of molecular codes allows to develop algorithms for code identification. In Chapter 4 I will present two algorithms, one based on closed

sets and one based on paths, to find all codes in a chemical reaction network and discuss the algorithms runtime properties. For the path based algorithm I propose two heuristic improvements, (1) by using the K-shortest paths, and (2) by a Monte-Carlo subnetwork sampling algorithm. In Chapter 5 I will present the results of the application of the algorithms to various biological and chemical systems. Chapter 6 discusses how the presented structural semantic level can be extended and validated by the pragmatic level. Finally, in Chapter 7 I will discuss further topics emerging from the presented formalism, algorithms, and results from actual networks. Appendix A contains a collection of algorithms and helper methods I used for the code identifying algorithms. Appendix B contains the detailed proof of the "ten closed sets" lemma applying to molecular codes. In Appendix C and D additional detail about results of a code based analysis of the human signal transduction network from the reactome database [2] and a metabolic network extracted from the KEGG database [3] are given, respectively. The network models of all analysed systems are collected in Appendix E.

[2] www.reactome.org

[3] www.genome.jp/kegg

Chapter 2

The notion of "Code" in biological research

Parts of this chapter have been published in [32].

Comparing the literature on codes in biological systems shows that the term "code" is used in two meanings, (1) as family of codewords, e.g. as in a block code, and (2) as mapping.

Both notions are used in recent biological literature, but not as formally defined as in information and coding theory (see Introduction). I will review three (major) biological systems that have been described to constitute molecular codes. I will discuss the used notion of code and give suggestions for a common usage of the term code as mappings.

2.1 Gene translation – The genetic code

The most prominent molecular code is the genetic code. In general the genetic code is referred to as the association between codons and amino acids. This is realised by amino acyl-tRNA synthetases (aaRSs) (for reviews on the genetic code see [17] and on aaRSs chemistry see [33]). There exist twenty different aminoacyl-tRNA synthetases[1], each one of them specific for one of the proteinogenic amino acids. A specific aaRSs realises a particular association between a tRNA and an amino acid. The specificity of the recognition is implemented mainly by interaction with the anticodon of the tRNA [33]. The anticodon is, as the codon on the DNA/mRNA, a codeword which can be described as an element of a block code of length 3, $GC_{Block} = \{A, C, G, T\}^3$. Thus, the tRNA/aaRSs system implements a reading system for this block code, i.e., the set of codewords. The semantic code is the decoding scheme consisting of the set of codewords $\{AAA, AAC, \ldots, TTT\}$ and the mapping from this set to the set of amino acid symbols $\{Ala, Gly, \ldots, Tyr\}$. The tRNAs function as adaptors of the code by realising two recognition processes (compare also [16]), i.e. between codon and tRNA and between amino acid and tRNA, and thereby realising the association between codon and amino acid.

The appealing feature of the genetic code is its simplicity. The coding table shows only the decoding function, i.e., the semantic aspect of the gene translation system. Such a simple description, that abstracts from the complex biochemical processes of recognition, would also be desirable for other molecular codes.

[1] Sometimes aaRSs are also called "codases" since they are the enzymes that implement the code [33, 34]

In a subsequent chapter (Chapter 5.5) the gene translation system will be analysed for its coding properties.

2.2 Covalent histone modifications - The histone code

Beside the genetic code other biological subsystems of the cell have been reported to constitute or contain codes [16]. In this section I will describe the system of histone modifications and discuss the possibility that it constitutes a molecular code.
In all kingdoms of life the DNA is organized in some kind of superstructure, a kind of packaging. This packaging is mainly maintained by so called "chromosomal architectural proteins" (chAPs), e.g., histones in eukaryotes. The existence of different modification sites on the tails of the histones led to the hypothesis that histone modifications could be part of a complex code, the histone code. At the moment there exist two theories how histone modifications can have an effect on gene regulation [35, 36]. The first one postulates a direct effect (in *cis*) of histone modifications on chromatin structure by altering the positive charge of the histone tails. The chromatin can regulate gene expression by its structure [37]. Dense chromatine inhibits transcription, while an open chromatine structure allows for transcription. The transcription in the latter case is possible because the DNA is accessible for the transcription machinery. Such an opening of the DNA at a histone can also be triggered by post-translational modifications of the histone tails. Certain modifications, like acetylations, can change the electrostatic properties of the protein-DNA interaction [38] and thus allow for an opening of the chromatin structure. This charge neutralisation weakens the interaction of histone tails and the DNA [38]. This theory applies only to acetylation and does not cover other types of modifications [35].
The second theory, the histone code hypothesis, has been introduced by Turner [39, 40], and Strahl and Allis [41]. It proposes that histone modifications are recognised and translated into biological functions [42] mediated by adaptor proteins (in *trans*) [43]
Talking about translation should refer to a decoding scheme, but from the definition and the usage of the term "code" in this context it is not quite clear what exactly "code" should mean here, the combinatorial patterns of modifications [44] or the mapping. In the former case the histone code would only be a family of code words.
From a semantic perspective the definition of a code must contain a mapping between the set of codewords and the set of encoded meanings. So in case of the histone code the codewords are modification patterns. But what are the meanings of the codewords, i.e., where are they mapped on? Different views have been reported, e.g., the modifications are mapped on (1) "downstream functions" [41], (2) "regulation of transcriptional activity" [45, 46, 47], (3) "other histone modification patterns"[35, 48].
In case of (1) the meanings could be high level functions, like meiosis, sporulation, etc. In case of (2) the meanings would basically be "on" and "off". And in case of (3) the meanings would be other patterns of histone modifications. Each of these three cases would constitute a different code.
It has also been proposed to use terms such as "language" and "grammar" in the case of histone cross-talk [36], but his does not contribute to a suitable description of the histone code as long as both terms are in need of a proper definition.
How could a histone code be realized by cells? Histone modifications can be actively written, read, and erased by protein domains [35, 36, 37]. (1) The combination of different reader domains in one protein or protein complex allows for the recognition of

not just single modifications, but patterns of modifications. This is for example the case for a tandem bromodomain reading two acetylated histone amino acids [49]. (2) The combination of reader domains and effectors (e.g., writing domains, erasing domains, or other enzyme functionality) allows for the coupling to biological function. Both features (1) and (2) together can make up the core of a histone code, because it makes the formation of adaptors possible. Therefore, by combining different domains, the cell would be able to read the codewords (patterns of modifications) of the histone code and relate them to some biological function. For proteins in general this has been referred to as "compositional semantics" [11]. An example for probable adaptors is the family of BAF complexes which contains several Bromo- (acetylation recognition), Chromo- (methylation recognition), and PHD-domains for combined modification recognition [50]. The meanings of the code then are given by the biological effects, or functions that are directly linked to the actions mediated by the adaptors. Other effects or behaviours, located downstream, may also depend indirectly on the histone code.

2.3 Glycan recognition – The sugar code

Another well-studied biological system has already been described in terms of code, i.e., the *sugar code* [51, 52, 53, 54]. Monosaccharids can by combined to glycans in various ways, resulting in an enormous amount of different glycans. The huge number of different combinations are supposed to be the code in the sugar code. Laine [55, 54] defined the coding capacity of the sugar code as number of combinations that can be formed with a fixed number of monosaccharids. E.g., $\approx 10^{15}$ different hexasaccharids can be formed from 20 monosaccharids. This notion of coding capacity is based on the idea that the combinations of different building block make up the code. But from a semantic point of view it is necessary to define the code also by referring to a mapping between two sets of molecular species. Then the number of different oligosaccharids alone does not constitute the coding capacity but is equal to the number of different possible codewords.

The sugar code, as a semantic concept, has also to refer to the lectins. Lectins are proteins which recognize glycans, i.e., they are reading domains. There are many lectins known in bacteria and viruses [56], plants [57], and animals [58] so that it can be hypothesized that sugar codes are ubiquitously distributed. For a semantic description of a possible sugar code I will present a simple abstract model of virus-cell recognition, which is based on some artificial assumptions. The model starts from the known fact that viruses uses lectins to recognise glycans, which are presented on the cell surface [59]. I here assume a system with two glycans (G1,G2), one species of cells (C1), two viruses (V1,V2), and two lectins (L1,L2). From an evolutionary perspective the cells can be combined with both sugars resulting in the cell-glycan combinations (C1G1,C1G2), while the viruses could evolve to utilise both lectins, resulting in (V1L1,V1L2,V2L1,V2L2). We assume here that the lectins are specific, such that lectin 1 may only bind to glycan 1, and lectin 2 only to glycan 2. Thereby we may also get all infection combinations of virus and cells (V1C1, V2C1). In such a system a code can be identified. It contains the decoding function between the combinations of cells and glycans (C1G1,C1G2) and the infected cells (V1C1,V2C1). The decoding function is realized by the virus-lectin combinations (V1L1,V2L2), which we could call "codemakers" following a suggestion of [31], or molecular contexts of the mapping. There exists an alternative set of combinations (V2L1,V1L2), i.e. context, realizing a different decoding function (see Figure 2.1).

In such a setting the combination of cell and glycan is a codeword for the infections that can occur. Important here is also that the meanings of the codewords are combinations of virus and cell (see Table 2.1).

Table 2.1 A possible (binary) sugar code. Here the C1-glycan combinations are the codewords, which are mapped by the molecular context onto the meanings, i.e. the infected cells.

Role	Molecular species
codewords	C1G1,C1G2
meanings	V1C1, V2C1
context	V1L1,V2L2
alt. context	V1L2, V2L1

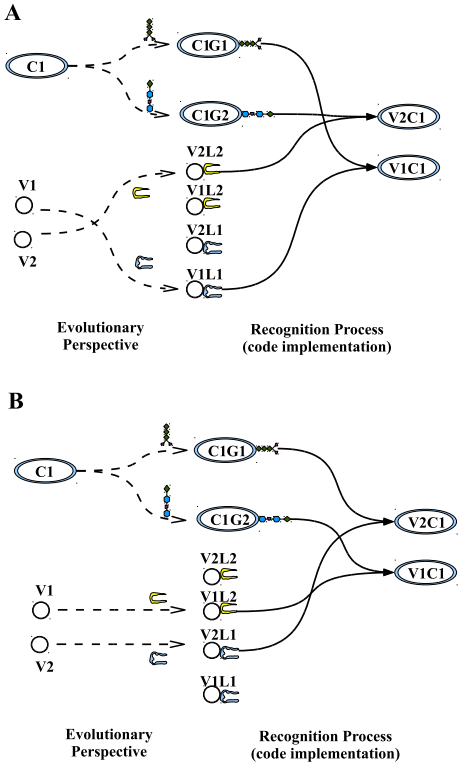


Figure 2.1 Model of a possible sugar code. Figures A and B show the realization of the two alternative mappings for the context and the alternative context. On the left hand side of A and B the evolutionary perspective indicates that both combinations between cells and sugars and virus and lectins should be possible in this scenario.(Reprinted from Publication BBA - General Subjects, Vol 1810(10), Dennis Görlich, Stefan Artmann, Peter Dittrich, Cells as semantic systems,914-923, Copyright (2011), with permission from Elsevier. Ref. [32])

2.4 Summary

The review of three systems discussed as codes in the literature shows that a proper formalised notion of codes is needed to foster that terms are used similarly. While for the genetic code it is commonly accepted that codons are mapped onto amino acids. For the other presented systems a clearer definition what the code is based on biological evidences would be also important. Best, the notion of code follows objective definitions. These are helpful to distinguish between the code, the code's execution, its evolution and pragmatics, the signs and the meanings in the code. Only the formal definition of code enables us to objectively discuss these in the various systems mentioned here.
The discussion of the biological systems also showed that the alphabet from which potential codewords are formed can be very heterogeneous. For example, to define the histone code's codewords the type of the covalent modification and its position is important, limited to the ability of the reading systems to recognize (complex) codewords.

Chapter 3

A formalisation of molecular codes

Parts and ideas of the contents presented in this chapter have been published in [60].
To access the notion of molecular codes for chemical and biological systems it is necessary to define it formally, best in a mathematical manner. This chapter introduces the formal framework for code based network analysis.

3.1 Formalisation of molecular codes in chemical reaction networks

Reaction networks are a suitable abstraction level to model systems of various kind. In the following I will define reaction networks (Def. 3.1.1), closed sets (Def. 3.1.4), paths (Defs. 3.1.2), because these concepts are important for the algorithmic identification of molecular codes.

Chemical reaction network Chemical reaction networks are usually defined by its molecular species, the reactions among these species and the kinetic laws governing the reactions (cf. [61]). For the definition of molecular codes I model only the static structure of a system as reaction network, such that the following definitions neglects kinetic information[1].

Definition 3.1.1 (reaction network)**.** *A chemical reaction network $N = (\mathcal{M}, \mathcal{R})$ is a tuple of a set of molecular species $\mathcal{M}$ and a set of reactions $\mathcal{R}$ given by $\mathcal{R} \subseteq \mathcal{P}(\mathcal{M}) \times \mathcal{P}(\mathcal{M})$ that can happen among the elements of $\mathcal{M}$. Each reaction $\rho \in \mathcal{R}$ is defined by its reactants $l_\rho \in \mathcal{P}(\mathcal{M})$ and products $r_\rho \in \mathcal{P}(\mathcal{M})$.*

Paths Intuitively, the molecular species of a reaction network N, eventually, are related by paths of reactions in the network. This allows to define relations among molecular species later on.

Definition 3.1.2 (s-t path)**.** *Given a reaction network $N = (\mathcal{M}, \mathcal{R})$ a* path *$p = (\rho_1, \rho_2, \ldots, \rho_i, \ldots, \rho_n)$ with $\rho_i \in \mathcal{R}$ is an ordered tuple of n reactions. In particular, the molecular species $s \in \mathcal{M}$ is called start species $s \in l_{\rho_1}$ and $t \in \mathcal{M}$ is called target species $t \in r_{\rho_n}$. For all sequential pairs of reactions $\rho_i, \rho_{i+1}, i \in \{1, 2, \ldots, n-1\}$ it should hold that at least one element of r_{ρ_i} is also in $l_{\rho_{i+1}}$:*

$$\forall i \in \{1, 2, \ldots, n-1\} : \exists m_i \in r_{\rho_i} \wedge m_i \in l_{\rho_{i+1}}.$$

[1] Kinetic information can be reintroduced later, e.g. for the pragmatic level, see Section 6

Corollary 3.1.1 (species s-t path)**.** *Each path in $N = (\mathcal{M}, \mathcal{R})$ induces a* species path *$p_t^s = (s, m_1, m_2, \ldots, m_i, \ldots, m_k, t)$ with $s, t, m_i \in \mathcal{M}$ as ordered tuple of $k+2$ species.*

Corollary 3.1.2. *A species path $p_t^s = (s, m_1, m_2, \ldots, m_j, \ldots, m_{n-2}, t)$ of length n induces a reaction path $p_{\rho_{n-1}}^{\rho_1}$ of length $n-1$, iff there exists $n-1$ reactions $\rho_i \in \mathcal{R}$, such that $s \in l_{\rho_1}, \quad t \in r_{\rho_{n-1}}, \quad m_j \in r_{\rho_j}, \quad m_j \in l_{\rho_{j+1}}$, with $j \in \{1, 2, \ldots, n-2\}$.*

Both notions of paths can be constructed from each other (Corollary 3.1.2), such that I will use the notion of *path* for the rest of this thesis and will refer to reactions or species as needed.

Molecular context In the following I will introduce the notion of the *molecular contexts* of a path. If a path from species s to species t does not only consist of spontaneous reactions a non-empty molecular context for this path can be identified. Following the reactions from s to t some of the reactants are produced by the preceding reactions, but some additional species may be necessary to execute all reactions among the path. I will call the set of these necessary molecular species "molecular context". In other words: The contexts consists of all molecular species that are not produced by a path, but necessary for the execution of the reactions.

Definition 3.1.3 (molecular context)**.** *Every s-t path induces a molecular context C which is necessary to execute the reactions on the path. For a path among species $(m_1, m_2, \ldots, m_n)$ and reactions $(\rho_1, \rho_2, \ldots, \rho_{n-1})$ the context is given by*

$$C = \bigcup_{i=1}^{n-1} l_{\rho_i} - m_i$$

For a given reaction network a particular path has only one context, because the path, by definition, has only one starting species and a defined set of reaction. The starting species and the set of reactions define the context.

Closed sets A useful concept to access the substructure of a reaction network is the notion of *closed sets* (cf. [62]). Intuitively, a closed set is set of molecular species that cannot produce "new" species that are not already contained in the set, thus, it stays closed.

Definition 3.1.4 (closed set)**.** *Given a reaction network $N = (\mathcal{M}, \mathcal{R})$ and a subset $A \in \mathcal{M}$ we say A is closed, iff for all reactions that can happen among the molecular species in A no new species are produced. If A is closed it holds that*

$$\forall \rho \in \mathcal{R} : l_\rho \subseteq A \rightarrow r_\rho \subseteq A.$$

The smallest closed set of an initial set A is called *closure* of A. The closure for any given set A can be calculated by the $G_{CL}()$ operator (Algorithm A.5). Algorithm A.3 gives the set of all closed sets CL_N.

A reaction network that contains the species A, B, C and one reaction, e.g. $A + B \to C$ contains two paths (A, C) and (B, C). The molecular context for path (A, C) is $\{B\}$ and the molecular context for path (B, C) is $\{A\}$. It also contains five closed sets $\text{CL} = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B, C\}\}$.

Definition 3.1.5 (single molecule closed set)**.** *Given a reaction network $N = (\mathcal{M}, \mathcal{R})$ the set of single molecule closed sets of N is defined as*

$$\text{SCL}_N = \{c \in \text{CL}_N | c = G_{CL}(m), m \in \mathcal{M}\}.$$

To define a molecular code I will start to define a molecular relation and a molecular mapping. In particular, a molecular code is a special case of a molecular mapping, which is a special case of a molecular relation.
The general definition of "relation", following [63], is:

Definition 3.1.6 (relation)**.** *Given two set A and B. A relation R is a subset of $A \times B$,*

$$R \subseteq A \times B. \tag{3.1}$$

For a reaction network N a relation R_N among the molecular species is given by $R_N \subseteq \mathcal{M} \times \mathcal{M}$.

Definition 3.1.7 (molecular mapping)**.** *Given a reaction network $N = (\mathcal{M}, \mathcal{R})$ and two sets of molecular species $A, B \subseteq \mathcal{M}$, we say that $f : A \overset{C}{\mapsto} B$ is a* molecular mapping *with respect to N, iff there exists a relation*

$$F = \{(a, b) \in A \times B | \text{a path } p = (a, \ldots, b) \text{ exists in } N\} \tag{3.2}$$

which is left-total $\forall a \in A \exists b \in B : (a, b) \in F$
and right-unique $\forall a \in A, b, c \in B : (a, b) \in F \wedge (a, c) \in F \to b = c$
with p realised by $C \subseteq \mathcal{M}$ (called context).

The left totality requires that all elements from the domain are used in the mapping, while right-uniqueness guarantees that no element of the domain maps to two elements from the codomain.
Alternatively closed set can be used to define a molecular mapping by defining

$$F = \{(a, b) \in A \times B | b \in G_{CL}(a \cup C)\}. \tag{3.3}$$

The calculation of the closure operator implies a repeated application of the operator to a set of molecular species. In each step the operator applies all possible reaction rules. By this the sequence of reactions leading to b is generated and also the s-t path. If there exists a molecular mapping f with respect to N, N can *realise* the molecular mapping f.
Note that in a reaction network there is usually more than one molecular context C that realises a particular molecular mapping f. Intuitively, in order to "compute" $f(a)$ with the reaction network N, we put all molecules from the context C together with a and repeatedly apply all applicable reaction rules until no novel molecular species can be added any more. Then it is checked which molecular species from the codomain B is

present, which must be – according to Definition 3.1.7 – only one species and the result of $f(a)$.
Based on the notion of a molecular mapping a molecular code can be defined. As outlined in the introduction, a code is a mapping between sets of objects, where the mapping could be different. To identify different mappings the alternative contexts needs to be identified.

Definition 3.1.8 (molecular code)**.** *Given a reaction network $N = (\mathcal{M}, \mathcal{R})$ and a non-constant*[2] *molecular mapping $f : A \overset{C}{\mapsto} B$, with $A, B, C \subseteq \mathcal{M}$ we call the mapping f a* molecular code *with respect to N, if all other mappings $g_i : A \overset{C'_i}{\mapsto} B$ with the same domain A and codomain B can also be realised by the reaction network N, i.e., there exist alternative molecular contexts C'_i to map A to B.*

The definition implements the notion of contingency, i.e. the elements of the domain can be mapped to the elements of the codomain in every possible way by changing the molecular context. Thus, networks that contain molecular codes realise an encoded relationship between molecular species by choosing or regulating a molecular context. Each code implies a family of potential molecular codes that are only distinguished by their molecular contexts. From these alternative mappings only few, perhaps only one, is realised in the systems that can be observed nowadays. If more than one of the alternative codes would be realised at the same time in the same system the mapping would not be right-unique, i.e. the mapping is no function any more.
The identification of a code, using our framework, does not guarantee that this particular code can be realised in the system. To finally verify a code's existence the pragmatic level needs to be added. On the pragmatic level the system has to choose, either by evolution, or by regulatory control, one of the alternative mappings to obtain a unique mapping (cf. Section 6). The identification of a code is a first measure if the (biochemical) system in principle could implement contingent mappings.

3.2 Binary molecular codes

In order to keep this study tractable, I will focus on molecular codes that are binary, i.e., where domain as well as codomain contain exactly two molecular species [60]. I will also not study molecular mappings that are only partially contingent. For binary molecular codes the definition can be reformulated as follows:

Definition 3.2.1 (binary molecular code)**.** *Given a reaction network $N = (\mathcal{M}, \mathcal{R})$ and two binary sets of molecular species $A = \{a_1, a_2\} \subseteq \mathcal{M}$ and $B = \{b_1, b_2\} \subseteq \mathcal{M}$. The molecular mapping $f : A \overset{C}{\mapsto} B$ is called* binary molecular code *(BMC), iff there exist two sets $C, C' \subseteq \mathcal{M}$, such that the following conditions hold:*

$$
\begin{aligned}
f(a_1) &\in G_{CL}(\{a_1\} \cup C), \text{ and } f(a_2) \notin G_{CL}(\{a_1\} \cup C), \text{ and} \\
f(a_2) &\in G_{CL}(\{a_2\} \cup C), \text{ and } f(a_1) \notin G_{CL}(\{a_2\} \cup C), \text{ and} \\
f(a_2) &\in G_{CL}(\{a_1\} \cup C'), \text{ and } f(a_1) \notin G_{CL}(\{a_1\} \cup C'), \text{ and} \\
f(a_1) &\in G_{CL}(\{a_2\} \cup C'), \text{ and } f(a_2) \notin G_{CL}(\{a_2\} \cup C').
\end{aligned}
$$

[2] A mapping $f : A \to B$ is called non-constant, iff there exists $a, a' \in A$ such that $f(a) \neq f(a')$.

Corollary 3.2.1 (code pair)**.** *A BMC always implies a code pair* $\mathcal{F} = (f, f', A, B, \mathcal{C}_f)$*, i.e. a tuple of the two alternative mappings, the domain, codomain and the joint contexts* $\mathcal{C}_f = \{C, C'\}$*.*

Two examples for reaction networks realising binary molecular codes are displayed in Fig. 3.1. Network A contains eight molecular species, four reactions and one code pair

$$\begin{aligned}\mathcal{CP}_A &= (\{(A1, B1), (A2, B2)\}, \{(A1, B2), (A2, B1)\}, \{A1, A2\}, \{B1, B2\}, \\ &\quad \{\{E1, E4\}, \{E2, E3\}\}).\end{aligned}$$

Network B contains six molecular species, four reactions and two code pairs

$$\begin{aligned}\mathcal{CP}_{B1} &= (\{(A1, B1), (A2, B2)\}, \{(A1, B2), (A2, B1)\}, \{A1, A2\}, \{B1, B2\}, \\ &\quad \{\{E1\}, \{E2\}\})\end{aligned}$$

and

$$\begin{aligned}\mathcal{CP}_{B2} &= (\{(E1, B1), (E2, B2)\}, \{(E1, B2), (E2, B1)\}, \{E1, E2\}, \{B1, B2\}, \\ &\quad \{\{A1\}, \{A2\}\}).\end{aligned}$$

The increased number of codes can be realised by the system, because $A1$, $A2$, $E1$, $E2$ can be used in two reactions equivalently (and symmetric) and thus can be exchanged as domain and context.

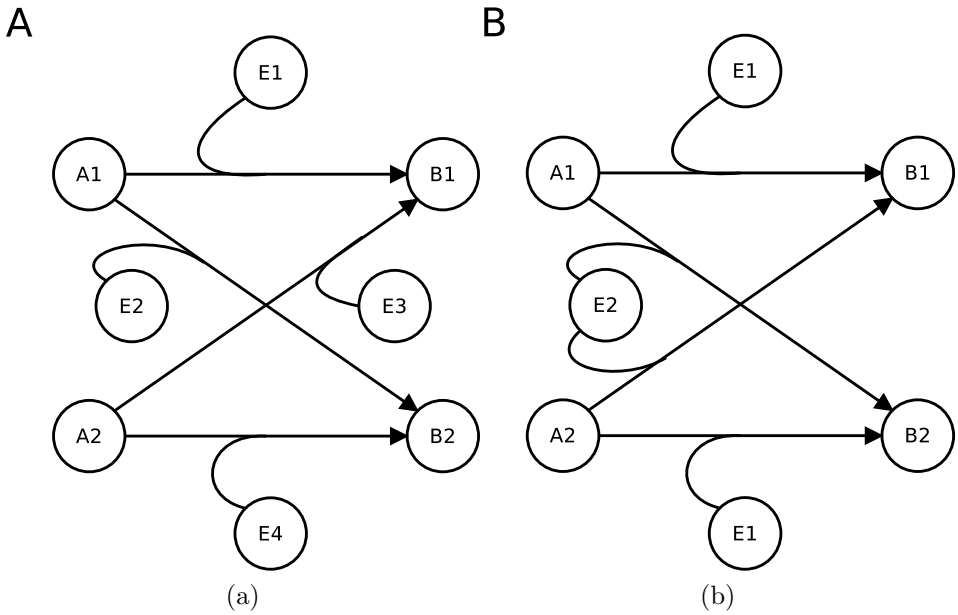


Figure 3.1 Example networks with binary molecular codes. A - The reaction network contains one molecular code pair mapping the set $\{A1, A2\}$ to $\{B1, B2\}$ either applying context $\{E1, E4\}$, or the context $\{E2, E3\}$. B - There are two code pairs that can be realised by this network. One mapping $\{A1, A2\}$ to $\{B1, B2\}$ using the context $\{E1\}$, or alternatively the context $\{E2\}$. The other code pair maps $\{E1, E2\}$ to $\{B1, B2\}$ using the context $\{A1\}$, or $\{A2\}$. The existence of the second code pair is due to the flexibility of the network, i.e., that $E1, E2$ and $A1, A2$ are capable to act in more than one reaction, such that they can exchange their role.

Lemma 3.2.1 (Ten unique closed sets)**.** *Given an BMC according to Definition 3.2.1 the ten closures* $G_{CL}(s_1), G_{CL}(s_2), G_{CL}(m_1), G_{CL}(m_2), G_{CL}(C), G_{CL}(C'), G_{CL}(s_1 \cup C) = G_{CL}(s_1 \cup C \cup m_1), G_{CL}(s_2 \cup C) = G_{CL}(s_2 \cup C \cup m_2), G_{CL}(s_1 \cup C') = G_{CL}(s_1 \cup C' \cup m_2)$, *and* $G_{CL}(s_2 \cup C') = G_{CL}(s_2 \cup C' \cup m_1)$ *must be different.*

If two of the above listed closed sets are not different the coding property vanishes, i.e. the signs or meanings get undistinguishable, or the relation is not unique because both meanings are generated at the same time. I call these situations *sign, or meaning degenerated*, respectively. A third form is that the contexts produce each other, i.e. the relation is *context degenerated*. For the proof by enumeration see Appendix B on page 117.
Lemma 3.2.1, leads to the conclusion that a network needs to be minimally structured in the sense that enough (> 10) different closed sets exists. This is, for example, not the case in a system where all the reactions happen spontaneously.

Lemma 3.2.2 (molecular code decomposition)**.** *Each molecular code* f *can be decomposed into* $\binom{|A|}{2} \cdot \binom{|B|}{2}$ *binary molecular codes.*

Proof. All molecular codes, following Definition 3.1.8, are completely contingent and thus each element of the domain can be mapped to each element of the codomain. By choosing two arbitrary elements from A and two arbitrary elements from B the result is always a BMC. Since there are $\binom{|A|}{2}$ pairs of elements in A and $\binom{|B|}{2}$ pairs of elements in B and each combination of these is a BMC. The product $\binom{|A|}{2} \cdot \binom{|B|}{2}$ gives the number of BMCs after decomposition. □

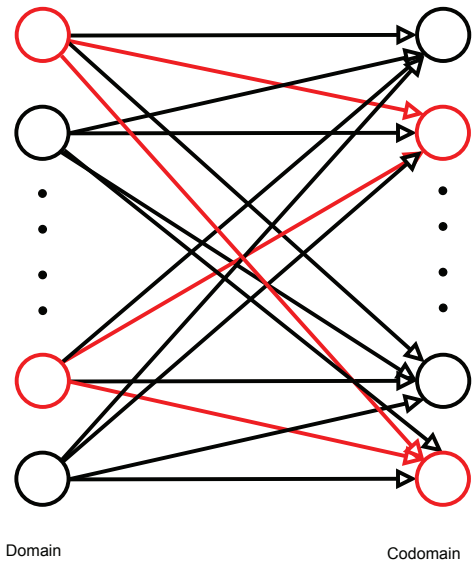


Figure 3.2 Decomposition of a molecular code into binary molecular codes. The figure shows a larger molecular code (only the mapping by omitting the molecular contexts). Each selection of two elements from the domain and two elements from the codomain results in a binary molecular code (indicated by the red coloured selection).

3.3 Semantic capacity

Biological systems seem to have a kind of semantic capacity, which allows them to evolve information processing systems. A system's semantic capacity, in general, can be defined

as capability to establish semantic relationships, i.e. to generate biological meaningful mappings. For the complete understanding of information processing, beside the pure syntactical description of signalling systems, the quantification of the semantic capacity is important. Very general properties of such a measure Sc of semantic capacity are:

- the measure should be non-negative, there is nothing like negative capacity
- monotonicity
- measured on a ratio scale (a non-arbitrary zero point)

As outlined in the introduction semantics is characterised by codes, thus it seems straight forward to measure the semantic capacity as number of (binary) molecular codes that can be realised by the system. Counting the number of binary molecular codes fulfils the properties stated above: The number of code pairs is non-negative, it grows in a monotonous way and it has no arbitrary zero.
In its basic form the semantic capacity is given by the number of codes pairs. Throughout this thesis I will apply this notion, but eventually, indicate potential modifications to this definition.

Definition 3.3.1 (semantic capacity)**.** *A system's semantic capacity Sc is its ability to realise contingent molecular mappings, i.e. the number of code pairs CP_N that can be identified in its reaction network model N, $Sc(N) = CP_N$.*

To compare large differences of semantic capacity the *logarithmic semantic capacity* can be used, defined as

$$Sc_{\log}(N) = \log_2(1 + Sc(N)) = \log_2(1 + CP_N)$$

especially with very high values of Sc. The transformation $1+x$ guarantees that $Sc_{\log}(N)$ is well defined and its smallest value is zero, in case the network cannot realise any molecular code.

3.4 Relations among codes

3.4.1 Code pair equality

For the analysis of real chemical networks it gets important to identify identical codes. I will present two definitions of code equality motivated by different aspects of the code, i.e. *structural* and *mapping* equality.

Definition 3.4.1 (structural code pair equality)**.** *Given two code pairs $\mathcal{F} = (f, f', A, B, \mathcal{C}_f)$ and $\mathcal{K} = (k, k', D, E, \mathcal{C}_k)$ $\mathcal{F} = \mathcal{K}$, iff*

$$\begin{aligned} f &= k \\ f' &= k' \\ A &= D \\ B &= E \\ \mathcal{C}_f &= \mathcal{C}_k. \end{aligned}$$

Two structurally equal codes are identical in all their components and thus are the same code.
From a functional perspective this may be a too strong constraint. In a biological system the exact composition of a code may be only one of many similar ways to implement a mapping. The mapping itself holds the functionality of the code. From this perspective the actual context is irrelevant and only the mapping can be used to identify identical codes.

Definition 3.4.2 (mapping code pair equality)**.** *Given two code pairs* $\mathcal{F} = (f, f', A, B, \mathcal{C}_f)$ *and* $\mathcal{K} = (k, k', D, E, \mathcal{C}_k)$ $\mathcal{F} =_m \mathcal{K}$*, iff*

$$\begin{aligned} f &= k \\ f' &= k' \\ A &= D \\ B &= E. \end{aligned}$$

The difference between the two definitions can be explained using the genetic code. Imagine two genetic codes $\mathcal{GC}_1$ and $\mathcal{GC}_2$. Both codes map codons onto amino acids using a set of tRNAs as context. The tRNA molecules are specific for codons and amino acids and determine the mapping. If both codes map the same codons to the same amino acids the both context consists of the same tRNAs and both codes are identical. If, for example, $\mathcal{GC}_2$ maps one codon differently the mapping and the contexts between both codes differ and thus two genetic codes would exist. This is true for both definitions. If, for example, both codes are identical in their mapping, but in $\mathcal{GC}_2$ a different pathway is used to map one of the codons to an amino acids (e.g. some post translational modification) compared to $\mathcal{GC}_1$. Then under Def. 3.4.1 both codes are different, while under Def 3.4.2 both would constitute one code.

3.4.2 Nested molecular codes

Molecular codes can be nested. A nested molecular code is "surrounded" by other molecular species that have incoming our outgoing reactions to the molecular code which leads to generation of (at least) a second molecular code (Figure 3.4). Such a configuration leads to an increased semantic capacity by combinatorics mediated by the nesting of molecular codes. Thus, a nested code can mediate a coded relationship between molecular species that are not directly involved in the code. Examples can be found in biology, e.g., in gene regulation. Here, the nested code is located at the DNA (see Section 5.6), while the observed encoded behaviour is between an external signal and internal states.
More formally, code nesting is a subset operation. The nested code relation is denoted by the $\Subset$ operator, with $\mathcal{F} \Subset \mathcal{K}$ if $\mathcal{F}$ is nested in $\mathcal{K}$, i.e. $\mathcal{F}$ is also called *core code pair*.

Definition 3.4.3 (nested molecular codes)**.** *Given the code pairs* $\mathcal{F} = (f, f', A, B, \mathcal{C}_f)$ *and* $\mathcal{K} = (k, k', D, E, \mathcal{C}_k)$ $\mathcal{F}$ *is included in* $\mathcal{K}$*, iff for* $c_f, c_{f'} \in \mathcal{C}_f, c_k, c_{k'} \in \mathcal{C}_k$

$$c_f \subseteq G_{CL}(D \cup c_k) \wedge c_{f'} \subseteq G_{CL}(D \cup c_{k'}) \tag{3.4}$$

$$\wedge \quad A \subseteq G_{CL}(D \cup c_k) \wedge A \subseteq G_{CL}(D \cup c_{k'}) \tag{3.5}$$

By the conditions in Def. 3.4.3 it is guaranteed, that if $\mathcal{F} \Subset \mathcal{K}$ then in $\mathcal{K}$ the reactions that realise $\mathcal{F}$ are used, i.e. $\mathcal{F}$ is completely contained in $\mathcal{K}$. This can either happen if $c_f \subseteq c_k$ or if the reactions among the outer code produce the domain and the context of the inner code. For Eq. (3.4) we can assume, without loss of generality, that the subsets of $\mathcal{C}_f$ and $\mathcal{C}_g$ are sorted, such that $c_f \subseteq c_g \wedge c_{f'} \subseteq c_{k'}$ is true.
Here I present which properties, e.g. reflexivity, are fulfilled by the nested code relation.

Lemma 3.4.1 (nested code reflexivity)**.** *Given a code pair $\mathcal{F} = (f, f'A, B, \mathcal{C}_f = \{c_f, c_{f'}\})$, then $\mathcal{F}$ is always its own core code, i.e. $\mathcal{F} \Subset \mathcal{F}$.*

Proof. For Eq. (3.4) we get $c_f \subseteq G_{CL}(D \cup c_f) \wedge c_{f'} \subseteq G_{CL}(D \cup c_{f'})$, which always hold for equality, since the G_{CL} operator does only increase the initial set. For Eq (3.5) we get $A \subseteq G_{CL}(A \cup c_f)$ which is by definition of G_{CL}always true using the same argument. Thus, $\mathcal{F} \Subset \mathcal{F}$ is always true. □

I continue by showing transitivity.

Lemma 3.4.2 (nested code transitivity)**.** *Given three molecular code pairs*

$$\begin{aligned} \mathcal{F} &= (f, f', A, B, \mathcal{C}_f = \{c_f, c_{f'}\}), \\ \mathcal{G} &= (g, g', D, E, \mathcal{C}_g = \{c_g, c_{g'}\}), \\ \text{and } \mathcal{H} &= (h, h', I, J, \mathcal{C}_h = \{c_h, c_{h'}\}) \end{aligned}$$

we say the binary relation $\Subset$ among $\mathcal{F}, \mathcal{G}$ and $\mathcal{H}$ is transitive if

$$\mathcal{F} \Subset \mathcal{G} \wedge \mathcal{G} \Subset \mathcal{H} \rightarrow \mathcal{F} \Subset \mathcal{H}. \tag{3.6}$$

I will only proof the lemma for one of the alternative molecular contexts. The proof for the second alternative is equivalent, but decreases readability, here.

Proof. We can directly proof this lemma using the equations from the definition 3.4.3. For Eq (3.4) we need to show that the following implications (which arises from (3.6)) hold.

$$c_f \subseteq G_{CL}(D \cup c_g) \wedge c_g \subseteq G_{CL}(I \cup c_h) \rightarrow c_f \subseteq G_{CL}(I \cup c_h) \tag{3.7}$$
$$A \subseteq G_{CL}(D \cup c_g) \wedge D \subseteq G_{CL}(I \cup c_h) \rightarrow A \subseteq G_{CL}(I \cup c_h) \tag{3.8}$$

To proof the implications we assume that the left hand sides of (3.7) and (3.8) are true and show that the right hand sides then also always are true. For Eq. (3.7) we know that $D, c_g \subseteq G_{CL}(I \cup c_g) = G_{CL}(I \cup c_g \cup D \cup c_g)$. Since the G_{CL}operator applies all possible reaction rules to the initial set $G_{CL}(D \cup c_g)$ is also a subset of $G_{CL}(I \cup c_g)$. Thus because of $A, c_f \subseteq G_{CL}(D \cup c_g)$ and $G_{CL}(D \cup c_h) \subseteq G_{CL}(I \cup c_h)$ we get

$$A, c_f \subseteq G_{CL}(D \cup c_g) \subseteq G_{CL}(I \cup c_h) \rightarrow A, c_f \subseteq G_{CL}(I \cup c_h)$$

which proofs, by standard set theory, Lemma 3.4.2. □

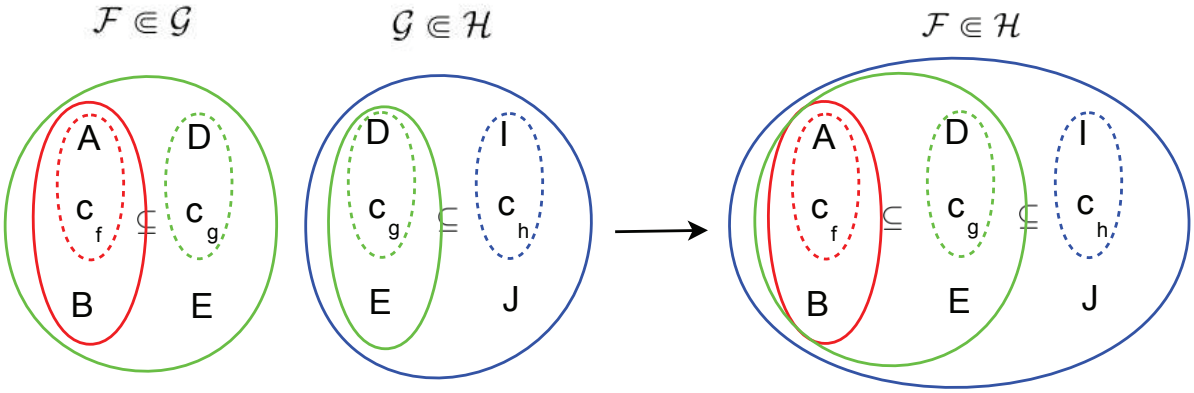


Figure 3.3 Subsets of transitive nested BMCs. The figure illustrates the proof for core code transitivity. On the left hand side the initial situation is displayed, i.e. $\mathcal{F}$ (red) is a core code of $\mathcal{G}$ (green) and $\mathcal{G}$ is nested in $\mathcal{H}$ (blue). Thus applying the closure operator to the initial subsets (dotted lines) generates a closed set containing the codomain of the larger code and the context and domain of the nested code, and therefore also the codomain of the nested code (solid coloured lines). Because all components of $\mathcal{F}$ are generated by $\mathcal{G}$ and all components of $\mathcal{G}$ are generated by $\mathcal{H}$, $\mathcal{F}$ is nested in $\mathcal{H}$.

The proof holds also for the alternative molecular codes.
So far I have proven that the core code relation is always reflexive and transitive.
The symmetry $\mathcal{F} \Subset \mathcal{G} \Rightarrow \mathcal{G} \Subset \mathcal{F}$ for $\mathcal{F} \neq \mathcal{G}$ is not valid in general. In actual networks there may be situations where symmetrically nested codes occur. This can happen if the molecular contexts are not identical, but share a core mechanism which realises the code. These codes are then very similar, if not the same code (reflexivity).
Also antisymmetry, $\mathcal{F} \Subset \mathcal{G} \wedge \mathcal{G} \Subset \mathcal{F} \rightarrow \mathcal{F} = \mathcal{G}$, is not always given for the nested code relation. This is due to the fact that the two code pairs can be nested, but their contexts may differ, such that the equality does not hold here.

Core code analysis of a toy network Figure 3.4 shows a reaction network containing a BMC motif surrounded by other molecular species only connected to the BMC motif by a simple incoming or outgoing reactions. In total the network contains 36 binary molecular codes. The codes reflects how one BMC motif can increase the semantic capacity by generating new mappings. These new mappings completely depend on the existence of the core code. Figure 3.5 illustrates the identified core code relations. Each node represents one of the BMCs. Each edge is directed to the core code, so $\mathcal{F} \Subset \mathcal{G}$ leads to an edge $F \rightarrow G$ in the graph. The size of each node represents the number of neighbours, while the color shows the connectivity. Each node has the reflexive edge. Transitivity can be best seen among the nodes 0,2,3 where 0 is a core code of 2 which is nested in 3. So, 0 is also nested in 3. The analysis of such core code relation networks allows to identify the generator codes, i.e. these induce many other secondary codes. Here, code number 0 has the maximal amount of neighbours which indicates that it is the generator of the complete semantic capacity. The measure of semantic capacity can be biased by the "generator effect" of core codes. An adapted measure should take into account the number of core codes. This is easy, if there exist only one nested code, but difficult if the relations between the identified codes are more complex. The core

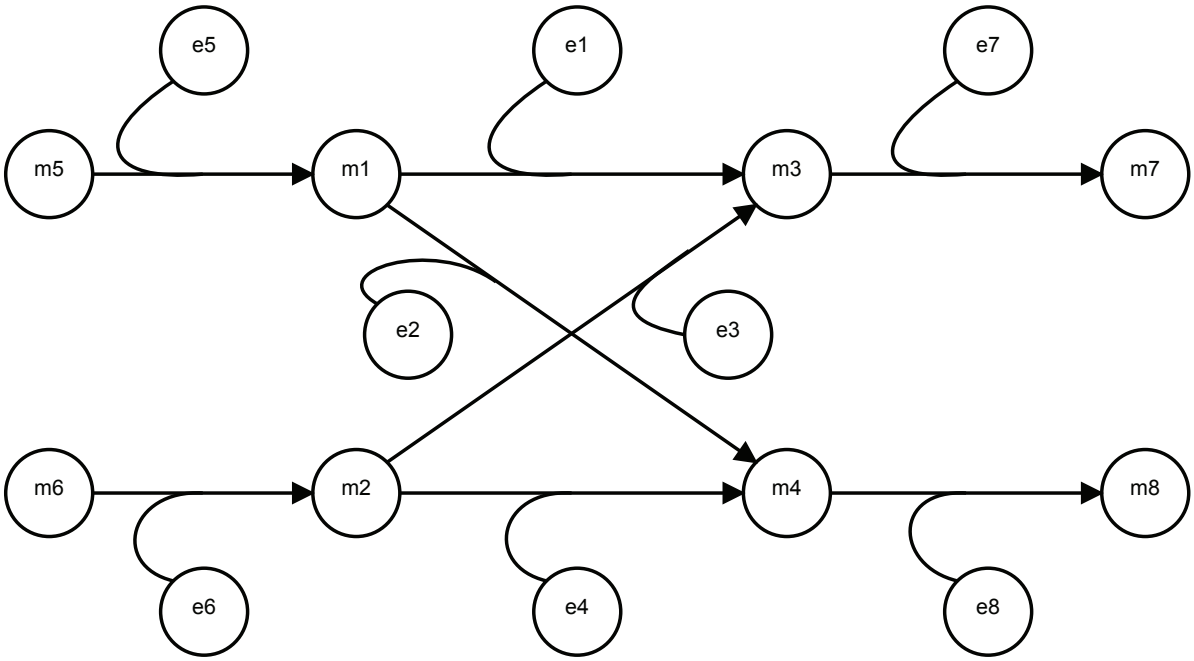


Figure 3.4 A reaction network with nested molecular codes. The reaction network contains 36 binary molecular codes. All of these can be reduced to the nested code $f : \{m1, m2\} \overset{\{e1,e2,e3,e4\}}{\longmapsto} \{m3, m4\}$. The code's nesting relation is shown in Figure 3.5.

core relations graphs may be structures in subgraphs, because there may exists different codes which are not in any relation. Given a set of code pairs we can calculate the semantic capacity as number of important core codes in each connected subgraph.

Definition 3.4.4 (semantic capacity by subgraphs)**.** *Given a reaction network N and its core code relation graph G we define the semantic capacity as number of unconnected subgraphs in G.*

Using this definition the measured semantic capacity will be reduced as soon as any two codes are in a core code relation.
From a pure structural point of view this reduction in semantic capacity describes the basic semantic capacity, since "pseudo" codes are not considered. From a pragmatic and biological point of view the other (induced) codes might also be relevant and thus important for the networks semantic capacity.

3.4.3 Code linkages

Molecular codes can show different degrees of dependencies. Code linkage is a concept that describes how two (or more) codes can be linked, such that the first (independent) code effects the execution of the dependent code. Code linkage can be observed in biological systems, e.g. in signal transduction where the signal transmission via the membrane (independent code) is linked to the gene regulatory code (dependent). The linkage is direct and realised by the second messengers and transcription factors. In the following I will define two types of code linkages.

Definition 3.4.5 (meaning-sign code linkage)**.** *Let $f : A \overset{C}{\longmapsto} B$ and $g : D \overset{C'}{\longmapsto} E$ be*

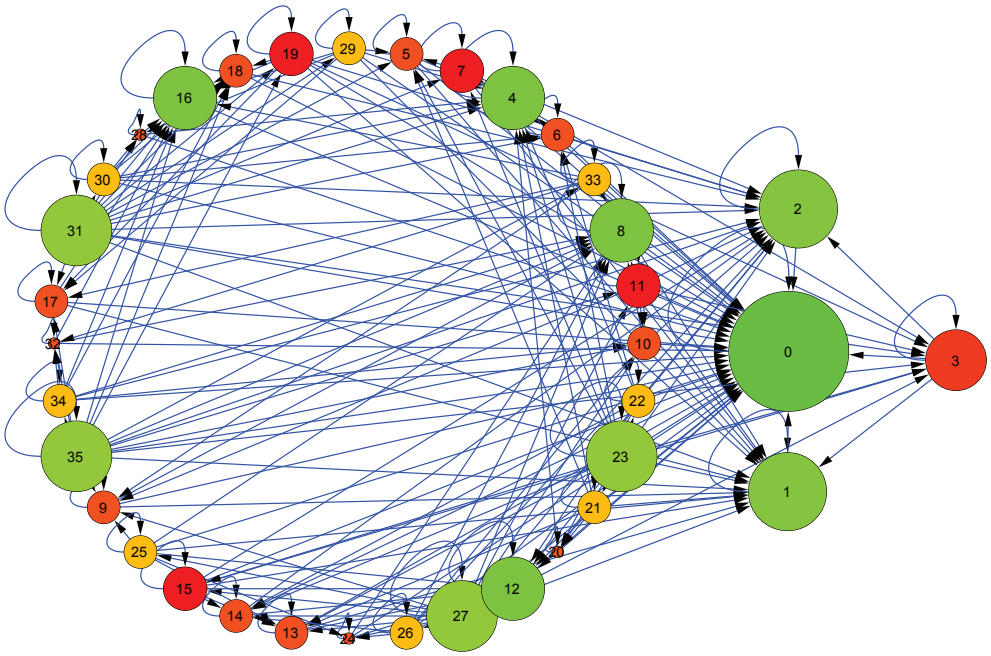


Figure 3.5 Core code relation network of Fig. 3.4. The graph shows which code is core code of which other code. There exists many core code relations. In particular, node number 0 is connected to every other node and thus is a kind of generator of all other codes. The size of the nodes represents the number of neighbors in the graph. Red nodes have a very high avg. connectivity, while green nodes have a lower avg. connectivity

molecular codes. g is linked directly to f, iff D is a subset of B.

$$g \prec_{MSCL} f : D \subseteq B$$

A meaning-sign code linkage (MSCL) can be observed for example in the gene regulatory system. Here the gene translation, i.e. the genetic code, is dependent on the output of the gene regulatory code (see section 5.6). The direct relationship comes into existence, because the output of the gene regulatory code, i.e. gene transcripts, is the input of the genetic code. Because the gene transcript is a sequence of codons the genetic code has to be executed several times, but in general they are directly linked.
MSCL increases the semantic capacity as measured by code pairs. Since through the linkage combinations of signs and meanings from f (which are signs from g) can be a molecular code mapping to the meanings of g. Figure 3.6 shows a MSCL situation, i.e. two linked BMC motifs. The network contains 23 BMCs.

Definition 3.4.6 (meaning-controlled molecular codes)**.** *Let $f : A \overset{C}{\longmapsto} B$ and $g : D \overset{C'}{\longmapsto} E$ be molecular codes. g is controlled by f, iff C' is a subset of B.*

$$g \prec_{MCMC} f : C' \subseteq B$$

Meaning controlled molecular codes (MCMC) describe the linkage, where the meaning of the first codes are elements of the molecular context of the second code and thus, by their presence, regulate the execution of the second code. Situations which might be governed by such a code linkage may be found in metabolic regulation. If a gene regulatory

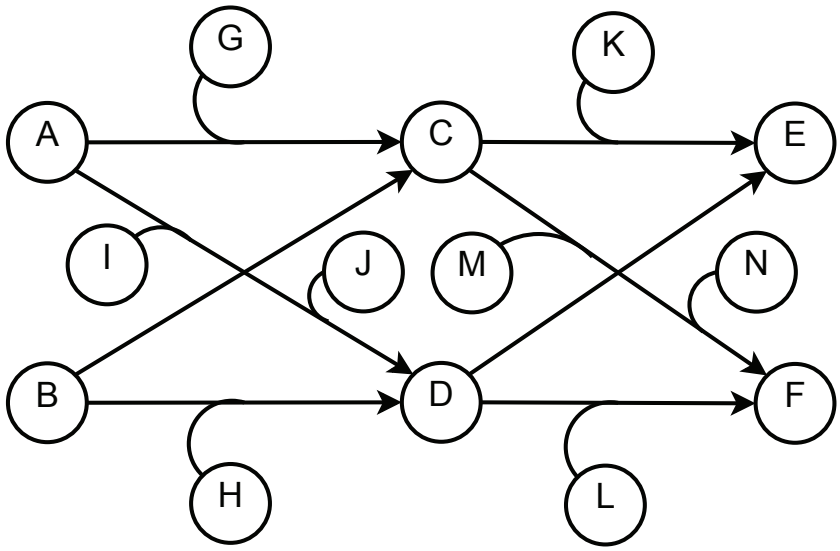


Figure 3.6 Example network for two linked BMCs via a meaning-sign code linkage (MSCL).

network, which can be considered as the first code (cf. Section 5.6), produces certain enzymes as meanings these may be part of a molecular context in a potential metabolic code. Then, the production of certain enzymes regulate the (encoded) production of certain metabolites.
Here, I described the first degree of code linkage, but the concept generalises to chains of codes. The signal transduction code governs the mapping of signals via the membrane, the secondary messengers are mapped on transcription factors which trigger the production of proteins that activate some effectors. Beside the activation of effector (which is a natural sign in some sense) all steps can be modelled as linked codes.

Chapter 4

Algorithmic code identification

Parts and first ideas of the contents presented in this chapter have been published in [60] and [32].
The formal definition of a binary molecular code (Def. 3.2.1) allows the formulation of suitable algorithms to identify BMCs in reaction networks. Here two algorithms for code identification are described, taking advantage of different properties of reaction networks, i.e. the number of closed sets and the pathways through the network. Both algorithms are directly derived from the definition of a BMC and follow a brute-force strategy by checking all combinations of either closed sets or molecular species for the code conditions.
Important for the successful identification of codes in reaction network model is that the model contains the alternative mappings. For several reasons network models available today does not contain all the alternative mappings. Before presenting the code identification algorithms I will discuss how suitable network models can be obtained.

4.1 Network representation

Today, network models of many biological systems are available from databases and can be downloaded in standardised formats like SBML [64, 65].
All formats have in common that the system's components needs to be represented in the network description. Network structure is mainly represented as list of molecular species and list of reactions among the species.
While modern file formats are mostly based on XML and thus contain also many annotations, e.g. kinetic information, I will here use a simplified network format, called REA-format, describing only the network structure. A rea-file (.rea) contains a list of molecular species, the number of molecular species, a list of reactions including stoichiometric coefficients and the number of reactions, in a plain text format. For compatibility the software can also use SBML Level 2 Version 1 files.

4.2 Obtaining suitable reaction networks

Classically, reaction networks are used to model actual biological or chemical systems. The network contains only the molecular species and reaction that have been observed before in the modelled system. Such networks, thus, can be called *realised* reaction networks. The set of realised reaction networks is a subset of all possible *potential* networks that could have been realised.

The notion of contingency used in the code definition given above, directly relates to potential reaction networks. The alternative molecular context characterises a potential different realisation of the mapping. Either the alternative mapping is present in the system, e.g. the system can switch between the mappings dynamically, or not, e.g. as in the genetic code. The latter case does not mean that no code exists, but that only one encoded mapping is fixed and the others are not realised (at the same time). To identify the molecular codes algorithmically it is necessary that all potential realisations of a code are present in one network model. This can be obtained by merging different networks, potential or realised ones, into one single reaction network.

A merge network can be constructed by a union operation:

Definition 4.2.1 (merge network)**.** *Given two reaction networks* $N_1 = (\mathcal{M}_1, \mathcal{R}_1)$ *and* $N_2 = (\mathcal{M}_2, \mathcal{R}_2)$ *we obtain a merge network* $N = (\mathcal{M}, \mathcal{R}) = N_1 \cup N_2$ *by*

$$\begin{aligned} \mathcal{M} &= \mathcal{M}_1 \cup \mathcal{M}_2 \\ \mathcal{R} &= \mathcal{R}_1 \cup \mathcal{R}_2. \end{aligned}$$

In particular, the merge operation implies that identical molecular species can be recognised and are present in the merge network only once. Also, merging network models from different environmental conditions may result in inconsistencies and "artificial" contingencies, e.g. if parts of a code can only be realised at completely different ranges of temperature, for example. For practical applications a network merge is a non-trivial task due to incomplete annotation of the networks.

Knowledge based network construction In some cases it is possible to construct a reaction network from expert knowledge. This works well if the modelled system is already well understood, but should not be applied in other cases. I used the knowledge based approach to analyse certain biological systems for their semantic capacity (see sections 5.5,5.6, and 5.8).

Once suitable network models are available the following algorithms for an automatic code identification can be applied.

4.3 Closure-based algorithm

The straight forward implementation of the BMC conditions (see Definition 3.2.1) leads to a closure based algorithm. The basic idea is to identify all BMCs by calculating all closed sets of the reaction network. Subsequently, every combination of six closed sets can be checked for the BMC conditions. In particular, for the domain and codomain only the single molecule closed sets are used (cf. Definition 3.1.5). Algorithm 4.1 shows the pseudocode of the closure based algorithm.

Algorithm 4.1 CLOSURECODEFINDER(N)

Input: A reaction network $N = (\mathcal{M}, \mathcal{R})$ with molecular species $\mathcal{M}$ and reactions $\mathcal{R}$.
Result: A list of code pairs consisting of a domain, codomain and two contexts.

```
1: clos ← ALLCLOSEDSETS(M)
2: SCL ← ∅
3: for all m ∈ M do
4:     SCL ← SCL ∪ {G_CL(m)}
5: end for
6: for all S_1, S_2, M_1, M_2 ∈ SCL do
7:     for all C, C' ∈ clos do
8:         if M_1 ⊆ G_CL(S_1 ∪ C) ∧ M_2 ⊈ G_CL(S_1 ∪ C)∧
           M_2 ⊆ G_CL(S_2 ∪ C) ∧ M_1 ⊈ G_CL(S_2 ∪ C)∧
           M_2 ⊆ G_CL(S_1 ∪ C') ∧ M_1 ⊈ G_CL(S_1 ∪ C')∧
           M_1 ⊆ G_CL(S_2 ∪ C') ∧ M_2 ⊈ G_CL(S_2 ∪ C')∧ then
9:             print (S_1, S_2, M_1, M_2, C, C')
10:        end if
11:    end for
12: end for
```

Helper methods:
G_{CL}see Algorithm A.5 on page 112, ALLCLOSEDSETSsee Algorithm A.3 on page 112.

The set of code pairs, resulting from the algorithm, depends on the used definition of code equality. For the counting of codes used in the definition of semantic capacity I used the mapping based definition of codes (Def. 3.4.2). The algorithm identifies different mappings, but ignores the context.

The runtime complexity of the closure based algorithm is mainly determined by the number of closed sets that have to be combined. Thus, the worst-case runtime complexity is bounded by $\mathcal{O}(|\text{SCL}|^4 \cdot n_c^2)$, with n_c as number of all closed sets. A closed term for the relation between closed sets and network size is not easy to develop, due to the strong dependency on the network structure. Intuitively, the less dense a network is, the more closed sets can be formed, but the actual relation between density and the number of closed sets needs to be investigated further.

4.4 Pathway-based algorithm

A second approach to implement a code-identifying algorithm can be realised by using the paths in the network model. Because the mapping between domain and codomain has to implemented by paths in the network model, the pathway based approach is equivalent to the closed-set approach. The resulting algorithm finds all BMCs in a reaction network with no prior information. The basic idea is to, first, calculate all s-t paths for all pairs of molecular species, and, second, check for every combination of four molecular species if they fulfil the conditions of Definition 3.2.1 by the paths connecting these four species.

Algorithm 4.2 pathCodeFinder(N)

Input: A reaction network $N = (\mathcal{M}, \mathcal{R})$ with molecular species $\mathcal{M}$ and reactions $\mathcal{R}$.
Result: A list of all code pairs the network can realise.

1: **for all** $s \in \mathcal{M}$ **do**
2: **for all** $t \in \mathcal{M}$ **do**
3: $paths^{st} \leftarrow$ getAllPaths(s, t)
4: **end for**
5: **end for**
6: **for all** $s, t, u, v \in \mathcal{M}$ **do**
7: **for all** $p^{st} \in paths^{st}$ **do**
8: **for all** $p^{uv} \in paths^{uv}$ **do**
9: **for all** $p^{sv} \in paths^{sv}$ **do**
10: **for all** $p^{ut} \in paths^{ut}$ **do**
11: $C_1 \leftarrow$ getContext$(p^{st}) \cup$ getContext(p^{uv})
12: $C_2 \leftarrow$ getContext$(p^{sv}) \cup$ getContext(p^{ut})
13: $Cl_{s,C_1} \leftarrow G_{CL}(\{s\} \cup C_1)$
14: $Cl_{u,C_1} \leftarrow G_{CL}(\{u\} \cup C_1)$
15: $Cl_{s,C_2} \leftarrow G_{CL}(\{s\} \cup C_2)$
16: $Cl_{u,C_2} \leftarrow G_{CL}(\{u\} \cup C_2)$
17: **if** $t \in Cl_{s,C_1} \wedge v \notin Cl_{s,C_1} \wedge t \notin Cl_{u,C_1} \wedge v \in Cl_{u,C_1} \wedge t \notin Cl_{s,C_2} \wedge v \in Cl_{s,C_2} \wedge t \in Cl_{u,C_2} \wedge v \notin Cl_{u,C_2} \wedge$ **then**
18: **print** (s, t, u, v, C_1, C_2)
19: **end if**
20: **end for**
21: **end for**
22: **end for**
23: **end for**
24: **end for**

getAllPaths has not been implemented.
Helper methods:
G_{CL} see Algorithm A.5 on page 112, getContext see Algorithm A.7 on page 113.

Theorem 4.4.1 (Completeness)**.** *Algorithm 4.2 finds all codes present in the network.*

Proof. All molecular codes are realised by the combination of paths between domain and codomain. Thus, if the algorithm considers all combinations of paths between all combinations of domains and codomain, i.e., checking all potential codes, it is guaranteed that all codes will be found. □

The path algorithm depends in its runtime complexity on the number of paths contained in the network. The number of paths is determined by the network size and density. Intuitively, the number of paths grows very fast with increasing network size. For example, the brute force algorithm for solving the travelling salesman problem has a runtime complexity of $\mathcal{O}(n!)$. The factorial determines the running time, because the algorithm basically enumerates all permutations of nodes in the graph, i.e. the potential paths. Similarly, the path based algorithm needs to check combinations of paths.

Theorem 4.4.2 (Runtime complexity path algorithm)**.** *For networks of size* $|\mathcal{M}|$ *and fixed density* d *the path based algorithm has a worst case runtime complexity of* $\mathcal{O}(|\mathcal{M}|!)$*.*

I will proof this theorem by applying results from the analysis of random networks published by Roberts and Kroese [66]. The authors basically presented an estimation on the number of s-t paths in random networks, verified by a Monte-Carlo sampling technique. I will use their result as estimate for the number of paths here.

Proof. Given a reaction network $N = (\mathcal{M}, \mathcal{R})$ of size $|\mathcal{M}|$ and density $d = \frac{|\mathcal{R}|}{(|\mathcal{M}|\cdot|\mathcal{M}|-1)}$ the number of s-t paths can be estimated by

$$\widetilde{\mathcal{Z}_{|\mathcal{M}|;d}} = K(|\mathcal{M}|) \cdot d^{|\mathcal{M}|-1+\delta(|\mathcal{M}|,d)},$$

where $K(n) = \sum_{k=0}^{n-2} \frac{(n-2)!}{k!}$, and $\delta(n,d) = \frac{3.32}{n} - \frac{5.16}{dn}$ [66]. The algorithm checks the BMC condition for all combinations of four molecular species $s, t, u, v \in \mathcal{M}$ by combining all paths for the combinations $(s,t), (u,v), (s,v), (u,t)$. For each combination of four species, $\binom{|\mathcal{M}|}{4}$, $\widetilde{\mathcal{Z}_{|\mathcal{M}|;d}}$ paths have to be checked at maximum leading to $\mathcal{O}(\widetilde{\mathcal{Z}_{|\mathcal{M}|;d}}^4 \cdot \binom{|\mathcal{M}|}{4})$. So the resulting algorithm solves the problem of identifying all binary molecular codes in polynomial time in the number of paths. Over network size, with a fixed density d, the factorial terms in $K(|\mathcal{M}|)$ dominate leading to $\mathcal{O}(|\mathcal{M}|!)$ as runtime complexity. □

The path algorithm (Algorithm 4.2) has very large running times (Theorem 4.4.2) at large networks and networks with many paths. By applying a parametrisation to the algorithm the runtime behaviour can be reduced. A straightforward parametrisation is to use only the K-shortest paths, instead of all paths, for every pair of molecular species as basis for the code identification. Identifying the K-shortest paths between two vertices of a graph is a general problem in graph theory for which several algorithms have already been developed [67, 68]. The K shortest paths problem has many important applications for finding alternative solutions in bioinformatics, e.g. metabolic pathway finding [69] problems.
In Algorithm 4.2 getAllPaths(s,t) is replaced by the function getKShortestPaths(s,t,K) leading to pathCodeFinder(N,K).
For getKShortestPaths(s,t,K) I use the freely available implementation[1] by Martin et al.[70]. The algorithm is based on Yen's algorithm [67] with a worst case running time of $\mathcal{O}(Kn(m + n\log n))$ to identify the K shortest paths between nodes s and t, with n as number of nodes and m as number of edges of the graph.
To use the implementation by Martin a preprocessing step is needed. The reaction network, which is mathematically a hypergraph, is transformed to a bipartite graph. The bipartite graph is generated by introducing a vertex for each reaction and by linking reactants and product to this vertex. A reaction `A + B → C + D` is transformed to `A → R, B → R, R → C, R → D`.
The graph used for the path identification then contains $|\mathcal{M}| + |\mathcal{R}|$ nodes. The number of edges m is given by the reaction's order and, thus, strongly depends on the network structure.

Theorem 4.4.3 (Runtime complexity for the K-shortest path algorithm)**.** *For networks of size $|\mathcal{M}|$ with fixed density d and a given K the K-shortest path based algorithm has a worst case runtime complexity of $\mathcal{O}(|\mathcal{M}|^4 K^4)$.*

Proof. As preprocessing step the K-shortest paths for all pairs of molecular species have to be calculated on the bipartite network model with size $N = |\mathcal{M}| + |\mathcal{R}|$. Because there

[1] Available at http://code.google.com/p/k-shortest-paths/

exist $\binom{|\mathcal{M}|}{2} = \frac{|\mathcal{M}|(|\mathcal{M}|-1)}{2}$ pairs of species, the runtime complexity of the preprocessing is

$$\mathcal{O}\left((KN(m + N\log N)\frac{|\mathcal{M}|(|\mathcal{M}|-1)}{2}\right).$$

Subsequently, for each combination of four species all combinations of the K paths has to be checked for the code property. Because for each combination of two species maximum K paths exist, the second part of the algorithms takes K^4 "time steps" per combination of four species. The second part of the algorithm is bounded by $\mathcal{O}(\binom{|\mathcal{M}|}{4}K^4)$. The runtime complexity of the complete algorithm (preprocessing + code checking) is the sum of the two terms leading to

$$\mathcal{O}\left(\left[(KN(m + N\log N)\frac{|\mathcal{M}|(|\mathcal{M}|-1)}{2})\right] + \binom{|\mathcal{M}|}{4}K^4\right).$$

The left term grows with a polynomial of order 2, while the right term grows with a polynomial of order 4 in $|\mathcal{M}|$, because $\binom{|\mathcal{M}|}{4} = \frac{|\mathcal{M}|(|\mathcal{M}|-1)(|\mathcal{M}|-2)(|\mathcal{M}|-3)}{4\cdot 3\cdot 2} = \frac{|\mathcal{M}|^4-6|\mathcal{M}|^3+11|\mathcal{M}|^2-6|\mathcal{M}|}{24}$. The polynomial of order 4 dominates the asymptotic runtime behaviour and for fixed K we get $\mathcal{O}(|\mathcal{M}|^4K^4)$ as final asymptotic runtime. □

The parametrisation bounded the factorial growth on paths and leaves a polynomial-time algorithm. The parametrised algorithm cannot find codes that use paths longer than the K shortest path. This can happen if many short paths exists between the potential sign and meaning that do not fulfil the code condition. This drawback can be eliminated by choosing K large enough, which results in larger running times, in the worst case again determined by a factorial. A promising result in this respect is that molecular codes are maintained to be efficient, i.e., their costs are minimised [23], so that it seems reasonable to assume that efficient molecular codes are realised by short paths. The parametrisation, thus, is likely not to miss the cost-optimal codes.

4.5 Implementation and runtime evaluation

The closure based and the K-shortest paths based algorithm have been implemented in Java.

I compared both algorithms for their practical runtime properties on different problem instances. As test networks I generated random reaction networks according to Algorithm A.2 with different size and density. Size and density have a direct effect of the number of closed sets and paths in the network. The more dense a network is the more s,t-paths between the species of the network exists. The number of closed sets decreases with growing density.

The closure algorithm is very quick on networks of size 5 and needs approximately the same amount of time for each network on average. This is a special case since these network does not have enough closed sets, where at least 10 closed sets are necessary for code identification (cp. Lemma 3.2.1). In general, the closure algorithm performs well on networks with higher densities (less closed sets) and worse on lower densities. For random networks of size 20 the running time is already very large ($> 1.7 \cdot 10^5 seconds \approx 2 days$).

The path algorithm shows the opposite behaviour. The more reactions are contained in a network the more paths needs to be checked, which increases the runtime. If K

Table 4.1 Empirical determined running times for the proposed algorithms measured by random test networks.

			closure algorithm	path algorithm	
$\|\mathcal{M}\|$	$\|\mathcal{R}\|$	N	mean runtime (s.e.m) in s	mean runtime (s.e.m) in s $K = 10$	mean runtime (s.e.m) in s $K = 20$
5	5	100	0.32 (0.004)	0.20 (0.00)	0.51 (0.04)
5	10	100	0.30 (0.004)	0.33 (0.01)	0.76 (0.04)
5	20	100	0.25 (0.003)	0.98 (0.02)	9.21 (0.41)
5	30	100	0.25 (0.002)	1.59 (0.03)	20.01 (0.62)
5	40	100	0.25 (0.003)	2.18 (0.04)	27.87 (0.81)
10	5	100	949.30 (33.30)	0.22 (0.00)	0.42 (0.03)
10	10	100	306.88 (15.05)	0.37 (0.01)	0.73 (0.03)
10	20	100	44.92 (3.45)	16.61 (0.74)	189.77 (12.20)
10	30	100	7.53 (0.70)	34.45 (0.81)	531.69 (13.32)
10	40	100	2.83 (0.24)	50.99 (0.94)	789.90 (15.27)
20	5	100	$> 1.7 \cdot 10^5$ (n.a.)	0.28 (0.00)	0.34 (0.002)
20	10	100	$> 1.7 \cdot 10^5$ (n.a.)	0.35 (0.01)	0.42 (0.007)
20	20	100	$> 1.7 \cdot 10^5$ (n.a.)	26.63 (3.43)	118.41 (25.69)
20	30	100	$> 1.7 \cdot 10^5$ (n.a.)	378.98 (18.55)	814.67 (109.25)
20	40	100	$> 1.7 \cdot 10^5$ (n.a.)	969.50 (22.45)	$> 1.7 \cdot 10^5$ (n.a.)

Run on an Intel(R) Core(TM)2 Duo CPU P8400 with 2.26 GHz and 2GB RAM.
Runtimes calculated by unix command `time -f "%E"`.

is increased, the runtime also increases because of the increased number of paths to be checked. As indicated by the values of the standard error the running times can vary a lot for a certain combination of size and density, because there may be single networks that, by chance, are easy to compute even if on average the computation is harder.

4.6 A random sampling algorithm for BMC identification

For large networks the identification of codes needs a large amount of time and computational resources. The theoretical runtime complexities (see above) suggest that for networks with either a large number of closed sets, or many paths the two algorithms may take long for a complete computation. Networks with a large number of closed sets, which are not feasible in the closure-based algorithm, contain only less paths and vice versa, such that the respective other algorithm can be applied, alternatively. Nevertheless, the data from the random network analysis (Section 5.1) suggests that networks with a large number of BMCs do have many closed sets and paths, such that the more interesting networks are likely infeasible for both algorithms. Assuming that a molecular code is realised mainly by shorter paths, codes could be identified in random subnetworks. By sampling random subnetworks there exists a remaining probability that some molecular codes are contained completely in a subnetwork, for example, if exactly the

subnetwork that is the code is sampled by chance. Algorithm 4.3 implements such a random subnetwork sampling with subsequent code identification. A subnetwork is sampled by randomly choosing (uniformly) an initial molecular species. Starting from this species the subnetwork is extended iteratively following Algorithm 4.4. In each step the network is extended by an incoming or outgoing reaction in an alternating manner. An incoming reaction is a reaction ρ where r_ρ is contained in the actual set of molecular species. In an outgoing reaction l_ρ is contained in the actual set of molecular species. The expansion algorithm stops when the number of molecular species is larger than a predefined threshold th_{size} (subnetwork size). The coverage parameter defines how many randomly sampled subnetworks are generated. The codes found in each subnetwork are collected, i.e. duplicates are removed and validated against the complete network. The validation step is necessary, because, due to the sampling, reactions not contained in the subnetwork (but in the original network) could destroy the coding property. The validation step (Algorithm 4.3, lines 6-10) is computational not expensive, it requires only the calculation of four closed sets (the combinations of two signs and two contexts) per code. The number of codes that can be identified with Algorithm 4.3 depends on the coverage and subnetwork size. To analyse the dependency on the three parameters subnetwork size, K, and coverage I use one of the networks analysed later in this thesis. The network (see Appendix E 5.2 on page 152) consists of 16 molecular species and 10 reactions and models a small gene regulatory network combined with the genetic code. The network contains 27 BMCs. I varied subnetwork size and coverage to show the effect of these two parameters on the rate of correctly found binary molecular codes (Figure 4.1). With growing subnetwork size the number of correctly identified codes increases. The data also clearly shows that under a certain critical subnetwork size (here, 10) no codes can be found even with growing coverage. Up to subnetwork size 15 the coverage also has only a small effect on the number of codes that can be identified. Only larger subnetworks and increased coverage yields better results. Overall, subnetwork size has the larger effect on the success of the algorithm, but also is increasing the computational effort. A trade-off exists between all three parameters and good settings need to be identified for each network model individually.

Algorithm 4.3 MonteCarloCodeSearch(N,n,K)

Input: A reaction network $N = (\mathcal{M}, \mathcal{R})$, an integer m, and integer n, and an integer K as parameter for the path algorithm.

Result: A list of binary molecular codes.

```
1:  candidates ← ∅
2:  for i = 0; i < n, i++ do
3:      N_sub ← EXPAND(N, m)
4:      candidates ← candidates ∪ PATHCODEFINDER(N_sub, K)
5:  end for
6:  for all C ∈ candidates do
7:      if C fulfils code conditions in N then
8:          print C
9:      end if
10: end for
```

The code finding algorithm PATHCODEFINDER is described in Algorithm 4.2 on page 40.

Algorithm 4.4 EXPAND(N,m)

Input: A reaction network $N = (\mathcal{M}, \mathcal{R})$.
Result: A subnetwork of N.

1: $\mathcal{M}_{sub} \leftarrow \emptyset$
2: $\mathcal{R}_{sub} \leftarrow \emptyset$
3: $initspec \leftarrow$ RANDOM$(0, |\mathcal{M}|)$
4: $\mathcal{M}_{sub} \leftarrow \mathcal{M}_{sub} \cup initspec$
5: **while** $|\mathcal{M}_{sub}| < m$ **do**
6: **if** $itermod2 == 1$ **then**
7: $r \leftarrow$ GETOUTGOINGREA$(\mathcal{M}_{sub}, N)$
8: **else**
9: $r \leftarrow$ GETINCOMINGREA$(\mathcal{M}_{sub}, N)$
10: **end if**
11: $reas \leftarrow$ GETREACTIONS(r)
12: $spec \leftarrow$ GETSPECIES$(reas)$
13: $\mathcal{M}_{sub} \leftarrow \mathcal{M}_{sub} \cup spec$
14: $\mathcal{R}_{sub} \leftarrow \mathcal{R}_{sub} \cup reas$
15: **end while**

Helper methods:
RANDOM() see Algorithm A.1 on page 111, GETOUTGOINGREA() see Algorithm A.8 on page 113, GETINCOMINGREA() see Algorithm A.9 on page 114, GETREACTIONS() see Algorithm A.11 on page 114, GETSPECIES() see Algorithm A.10 on page 114.

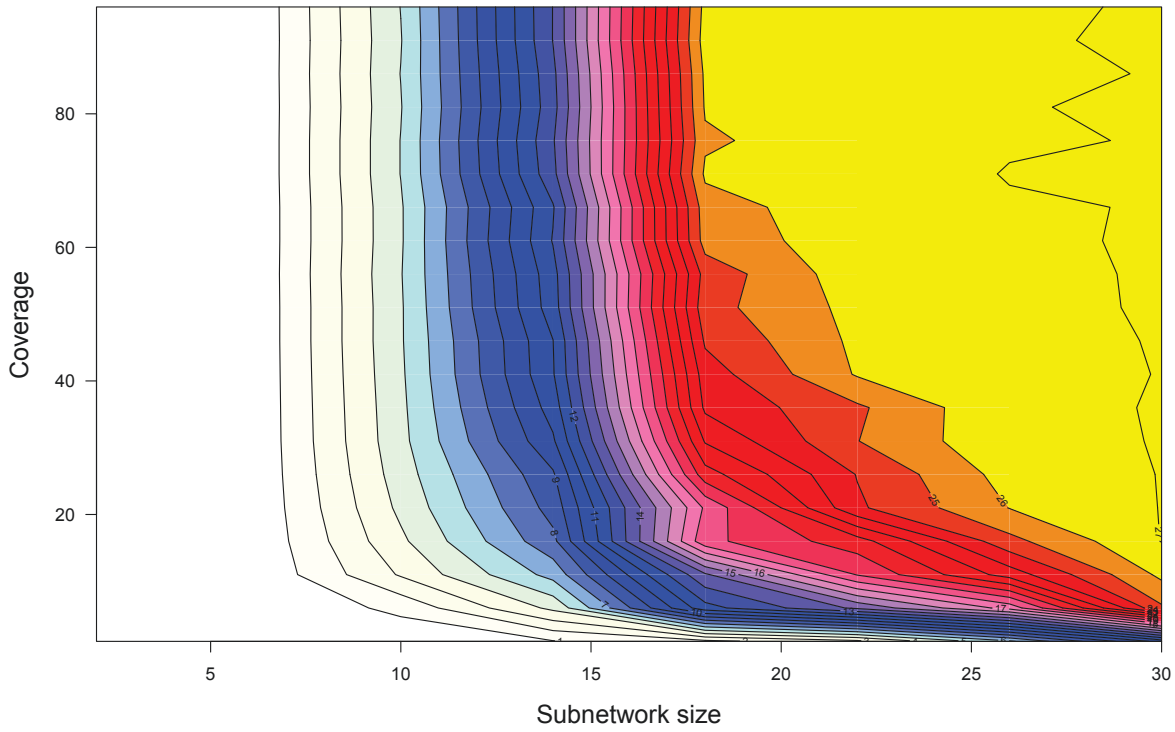


Figure 4.1 Results of the parameter scan for the random subnetwork sampling algorithm. For varied subnetwork size and coverage the plot shows that number of BMCs identified in the GC-GRN network (Appendix E 5.2). Color range from white (0 = no codes) to yellow (27 = max number BMCs).

4.7 Code completion

In many cases the knowledge about the system is insufficient to generate a complete network model. In principle, it can be assumed that most of the biological network models have missing reactions, or interactions not discovered, yet. They are an incomplete model of reality. For the code analysis this is a huge drawback since one missing edge is sufficient to prevent the identification of a code.
There are two ways to estimate how many incomplete code patterns are in a reaction network:

- Construct a new network model, by inserting an edge between an arbitrary pair of molecular species and rerun the code identifying algorithm

- Reformulate the BMC definition to a partial form in which one edge is missing and run the modified algorithm on the original network

From a computational point of view the latter option is favoured since its not increasing the runtime complexity and only needs one further analysis of the network (while the first option requires $|\mathcal{M}| \cdot (|\mathcal{M}| - 1)$ additional runs).

Definition 4.7.1 (incomplete binary molecular code)**.** *Given a reaction network* $N = (\mathcal{M}, \mathcal{R})$ *and two binary sets of molecular species* $A = \{a_1, a_2\} \subseteq \mathcal{M}$ *and* $B = \{b_1, b_2\} \subseteq \mathcal{M}$*. The molecular mapping* $f : A \overset{C}{\mapsto} B$ *is called an* incomplete *binary molecular code , iff there exist two sets* $C, C' \subseteq \mathcal{M}$*, such that the following conditions hold:*

$$
\begin{aligned}
f(a_1) &\notin G_{CL}(\{a_1\} \cup C), \text{ and } f(a_2) \notin G_{CL}(\{a_1\} \cup C), \text{ and} \\
f(a_2) &\in G_{CL}(\{a_2\} \cup C), \text{ and } f(a_1) \notin G_{CL}(\{a_2\} \cup C), \text{ and} \\
f(a_2) &\in G_{CL}(\{a_1\} \cup C'), \text{ and } f(a_1) \notin G_{CL}(\{a_1\} \cup C'), \text{ and} \\
f(a_1) &\in G_{CL}(\{a_2\} \cup C'), \text{ and } f(a_2) \notin G_{CL}(\{a_2\} \cup C').
\end{aligned}
$$

Definition 4.7.1 is illustrated by Figure 4.2. Instead of just leaving away one of the conditions one of the paths from domain to codomain is explicitly forbidden. The identification of this pattern can be reformulated as the question of which reaction needs to be included in the network to allow for coding between domain A and codomain B. A more reduced BMC pattern, that could cope with more inconsistencies and the incompleteness of a network model, allows to artificially generate contingent mappings and is not applicable. The same is true for an iterated, sequential introduction of the code completing edges.

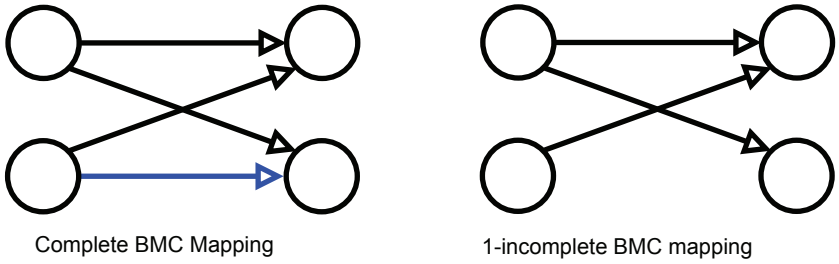


Figure 4.2 Comparison of complete and incomplete BMC. By directly disallowing one edge in the BMC condition I search for mappings as displayed on the right side. By inserting this edge (blue) in the network a complete BMC can be reestablished.

For the example network shown in Figure 4.3 the application of the code completion algorithm predicts, that four new code pairs could be realised by the system, by inserting the corresponding reactions(see Table 4.2). By structure these four codes are very similar and arise from the symmetry of the network. Figure 4.3 illustrates one of the predicted BMCs.
Applied on a network with an incomplete BMC pattern, i.e. one reaction is missing (Figure 4.4), the algorithm shows that the BMC can be restored, as expected. Additionally, a second potential code is found.

Table 4.2 Table of the predicted BMCs in the simple BMC reaction network using the code completion algorithm.

Domain	Codomain	Context	predicted reaction
$\{E1, A2\}$	$\{B1, B2\}$	$\{A1, E3, E4, E\star\}$	$E1 + E\star \rightarrow B2$
$\{E2, A2\}$	$\{B1, B2\}$	$\{A1, E3, E4, E\star\}$	$E2 + E\star \rightarrow B1$
$\{E3, A1\}$	$\{B1, B2\}$	$\{A2, E1, E2, E\star\}$	$E3 + E\star \rightarrow B2$
$\{E4, A1\}$	$\{B1, B2\}$	$\{A2, E1, E2, E\star\}$	$E4 + E\star \rightarrow B1$

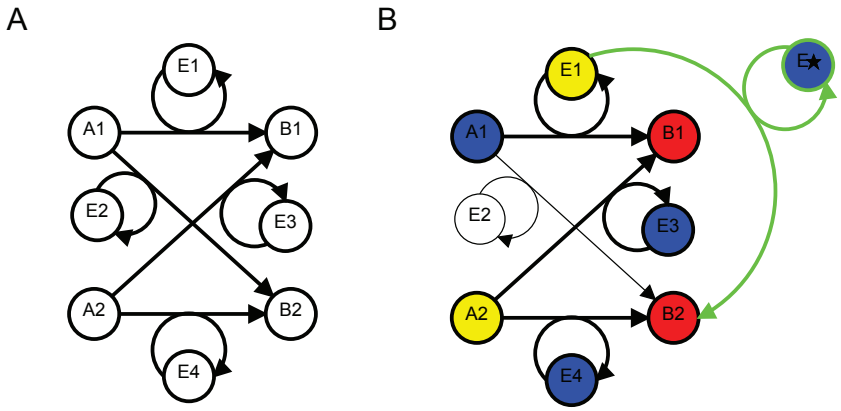


Figure 4.3 Result of the code completion algorithm on the complete BMC network. By applying the algorithm for code completion on the BMC network it can be seen that the network (panel A) is able to realise more codes by insertion of new reactions. Because of the symmetry of the network, here, four new code pairs could be implemented. Panel B shows one of these new code pairs, all four are listed in Table 4.2. yellow – domain; red – codomain; blue – context; green – newly inserted reaction

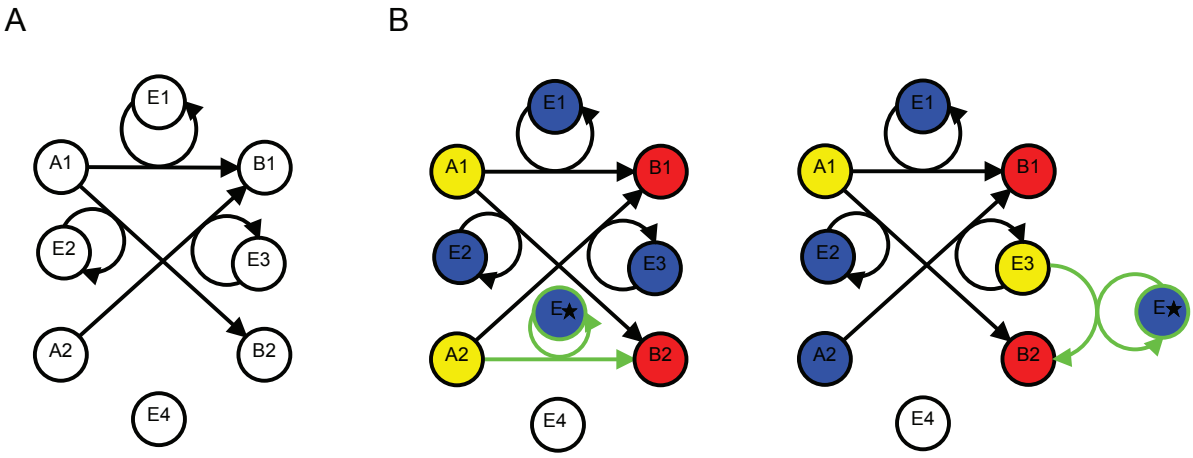


Figure 4.4 Result of the code completion algorithm on an incomplete BMC network. By applying the algorithm for code completion on the incomplete network the BMC can be restored by including the missing reaction $A2 + E\star \rightarrow B2$. Here, a second potential BMC comes up, if the reaction $E3 + E\star \rightarrow B2$ is inserted in the network model. Panel B shows the two new code pairs. yellow – domain; red – codomain; blue – context; green – newly inserted reaction

Chapter 5

Results of the algorithmic code analysis of various systems

Parts and first ideas of this chapter have been published in [60] and [32].
In this chapter I present the algorithmic, code based analysis of a number different networks, among them random reaction networks, combustion chemistries, gene translation, gene regulation, protein assembly networks and an artificial chemistry. Finally, I will present the results on two large scale biological networks and discuss problems in the analysis that can arise using the algorithmic code identification on database derived networks.

5.1 Random networks

I analysed random networks for their capability to realize binary molecular codes. The static definition of molecular codes results in a combination of molecular species and paths and thus the probability, that such a pattern occurs by chance, is larger than zero. The probability depends on three factors:

- network size – if the network is not large enough, the code pattern can not be generated
- network density – if there does not exist enough connection/reactions between the molecular species the paths between domain and codomain can not established
- reaction order – to establish a molecular context reactions of (at least) order 2 are needed. A network with only spontaneous reactions can not have molecular codes

For this study I generated random networks of varying size and density, but with a fixed reaction order. Random reactions are of the form $A + B \rightarrow C$, i.e. each reaction is "regulated" in the sense that a second molecular species is necessary for the reaction. Algorithm A.2 describes the network generation. In principle, it is possible to vary also the distribution of reaction orders in the networks. For this study I am primarily interested in size and density, because these two parameters directly influence the number of paths and closed sets (cf. the formulation of the algorithms, Chapter 4). Reaction order plays, therefore, only a minor role and is kept constant. For each combination of network size and density I generated 1000 random networks and applied the code identifying algorithm.

The number of code pairs in random networks follows a unimodal distribution. Figure 5.1 shows the results of the analysis of the random networks. In general, it can be observed that the number of paths increases with increasing density (compare runtime complexity of the algorithm in Section 4.4). The number of closed sets decreases with increasing density.

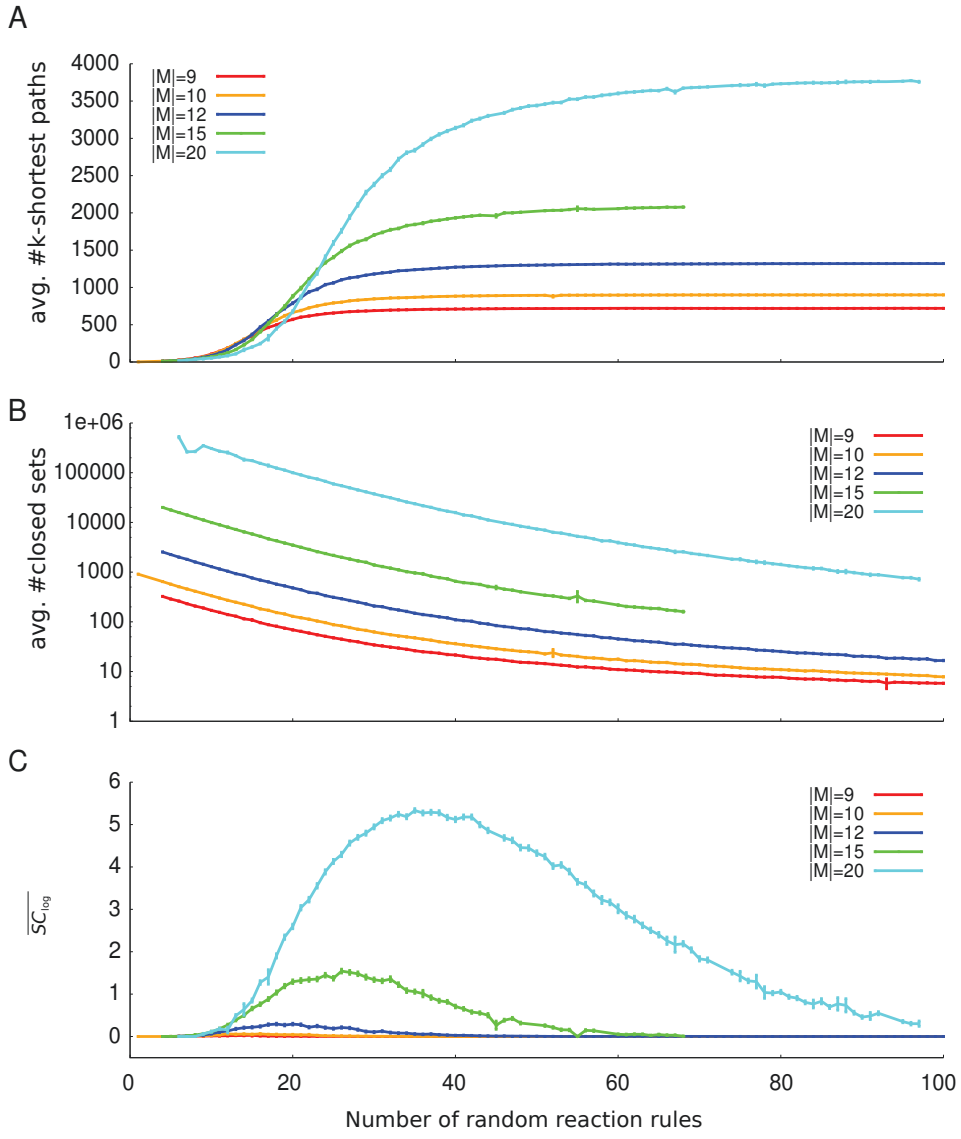


Figure 5.1 Code based analysis of random networks. Panel A shows the mean number of paths. The number of paths reaches a plateau because of the parametrization of the algorithm (K=10) . Panel B shows the mean number of closed sets and Panel C shows the average (log) semantic capacity over density of the generated random networks (N=1000). Error bars show the standard error of the mean.

The result of the code based analysis shows that random reaction networks in principle are capable of realizing binary molecular codes. What can be observed is that

- over density (for a fixed network size) the number of codes show a unimodal distribution,
- the maximum number of codes increases exponentially with network size,

- the position of the mean (and thus the position of the optimal interval) shifts linearly to larger densities with network size.

The extend of the distribution (Figure 5.1C) gives an optimal interval for random code generation, i.e., random networks with this size and a density lying in the interval are very likely to have codes by chance.

Statistical (null-)model. For the development of a null-model that allows the prediction of the semantic capacity also for combinations of network sizes and densities that have not been generated as random networks I developed a statistical model.
To obtain such a statistical model I assume that the average number of code pairs follows an unknown probability distribution and fit a statistical model on the data.
In general, the mean number x_s of BMCs over the network density for a fixed network size is modelled as random variable $\mathcal{X} \sim \mathcal{D}$. $\mathcal{X}$ follows an unknown probability distribution $\mathcal{D}$.
As candidate distributions I chose the normal ($\mathcal{N}(\mu, \sigma^2)$), the log - normal ($\ln\mathcal{N}(\mu, \sigma^2)$) and a gamma distribution ($\Gamma(k, \theta)$)[1]. All show a unimodal behavior for certain parameter combinations, but behave differently in their properties (e.g. skewness). All three distributions are commonly used for statistical purposes.
My approach here will be to estimate the candidate distribution's parameters from the data by using the empirical mean $\hat{\mu}$ and variance $\hat{\sigma}^2$. I calculate the goodness of fit to select the most suitable model.
In the following I show how the candidate distribution's parameters are related to the empirical mean and variance.

Normal distribution The normal distribution's probability density function is given by

$$f_{\mathcal{N}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}.$$

The normal distribution's mean and variance are given by μ and σ^2, such that for the estimate the empirical values can be used directly.

Log-normal distribution The log-normal distribution is a probability distribution whose logarithm is normally distributed. The probability density function is given by

$$f_{\ln\mathcal{N}}(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{\frac{-(\ln x-\mu)^2}{2\sigma^2}}.$$

The mean of the distribution is given by $e^{\mu+\frac{\sigma^2}{2}}$, while the variance is given by $(e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$. To calculate the distribution's parameters from the empirical mean and variance I will solve the following system of equations for μ and σ^2:

$$\hat{\mu} = e^{\mu+\frac{\sigma^2}{2}} \tag{5.1}$$
$$\hat{\sigma}^2 = (e^{\sigma^2} - 1)e^{2\mu+\sigma^2}. \tag{5.2}$$

I solve Eq. (5.1) for μ.

[1]Not to be confused with the gamma function, which is defined via factorials, but is used to calculate the gamma distribution.

$$\begin{aligned} \hat{\mu} &= e^{\mu+\frac{\sigma^2}{2}} \\ \Leftrightarrow \log\hat{\mu} &= \mu + \frac{\sigma^2}{2} \\ \Leftrightarrow \mu &= \log\hat{\mu} - \frac{\sigma^2}{2} \end{aligned} \tag{5.3}$$

Now I solve the Eq.(5.2) for μ and obtain

$$\begin{aligned} \hat{\sigma}^2 &= (e^{\sigma^2}-1)e^{2\mu+\sigma^2} \\ \Leftrightarrow \log\hat{\sigma}^2 &= \log(e^{\sigma^2}-1) + 2\mu + \sigma^2 \\ \Leftrightarrow 2\mu &= \log\hat{\sigma}^2 - \log(e^{\sigma^2}-1) - \sigma^2 \\ \mu &= \frac{1}{2}\left(\log\hat{\sigma}^2 - \log(e^{\sigma^2}-1) - \sigma^2\right) \end{aligned} \tag{5.4}$$

By equating Eqs. (5.3) and (5.4) the relation between the empirical estimates and σ^2 is obtained.

$$\begin{aligned} \log\hat{\mu} - \frac{\sigma^2}{2} &= \frac{1}{2}\left(\log\hat{\sigma}^2 - \log(e^{\sigma^2}-1) - \sigma^2\right) \\ \log\hat{\mu} &= \frac{1}{2}\left(\log\hat{\sigma}^2 - \log(e^{\sigma^2}-1) - \sigma^2\right) + \frac{\sigma^2}{2} \\ \log\hat{\mu} &= \frac{1}{2}\left(\log\hat{\sigma}^2 - \log(e^{\sigma^2}-1)\right) \\ 2\log\hat{\mu} &= \log\hat{\sigma}^2 - \log(e^{\sigma^2}-1) \\ -2\log\hat{\mu} + \log\hat{\sigma}^2 &= \log(e^{\sigma^2}-1) \\ \log\frac{\hat{\sigma}^2}{\hat{\mu}^2} &= \log(e^{\sigma^2}-1) \\ \frac{\hat{\sigma}^2}{\hat{\mu}^2} &= e^{\sigma^2}-1 \\ 1+\frac{\hat{\sigma}^2}{\hat{\mu}^2} &= e^{\sigma^2} \\ \log\left(1+\frac{\hat{\sigma}^2}{\hat{\mu}^2}\right) &= \sigma^2 \end{aligned} \tag{5.5}$$

I can use the solution for σ^2 (Eq. (5.5)) in Eq. (5.3) to get the relation for μ by

$$\begin{aligned} \mu &= \log\hat{\mu} - \frac{\log\left(1+\frac{\hat{\sigma}^2}{\hat{\mu}^2}\right)}{2} \\ \mu &= \log\hat{\mu} - \log\left(\sqrt{1+\frac{\hat{\sigma}^2}{\hat{\mu}^2}}\right) \\ \mu &= \log\left(\frac{\hat{\mu}}{\sqrt{1+\frac{\hat{\sigma}^2}{\hat{\mu}^2}}}\right) \end{aligned} \tag{5.6}$$

Gamma distribution The gamma distribution is given by the probability density function

$$\Gamma(k,\theta) = \frac{1}{\theta^k \Gamma(k)} \cdot x^{k-1} \cdot e^{-\frac{x}{\theta}}.$$

By definition the mean and the variance of a gamma distribution are $k\theta$ and $k\theta^2$, respectively.
To calculate k and θ from the empirical mean and variance I solve

$$\begin{aligned} \hat{\mu} &= k\theta \Leftrightarrow k = \frac{\hat{\mu}}{\theta} \\ \hat{\sigma}^2 &= k\theta^2 \Leftrightarrow k = \frac{\hat{\sigma^2}}{\theta^2}, \end{aligned}$$

by equating the two terms and obtain

$$\frac{\hat{\mu}}{\theta} = \frac{\hat{\sigma}^2}{\theta^2} \Leftrightarrow \theta = \frac{\hat{\sigma}^2}{\hat{\mu}} \tag{5.7}$$

$$k = \frac{\hat{\mu}}{\frac{\hat{\sigma}^2}{\hat{\mu}}} \Leftrightarrow k = \frac{\hat{\mu}^2}{\hat{\sigma}^2}. \tag{5.8}$$

Fitting the model. To obtain an estimate for arbitrary values of size and density I also modelled the behaviour of the empirical mean and variance of the unimodal distributions of BMCs.
The means of the unimodal distributions increases linearly (see Figure 5.2) with the increasing network size. For the variance the linear model does not fit well, such that I use an exponential model (see Figure 5.3).

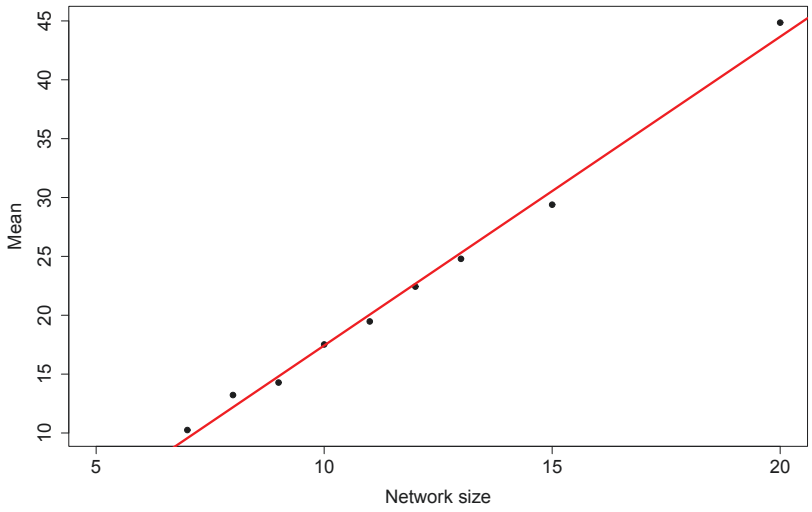


Figure 5.2 Mean number of reactions of the empirical unimodal distributions over size. Linear regression see Table 5.1.

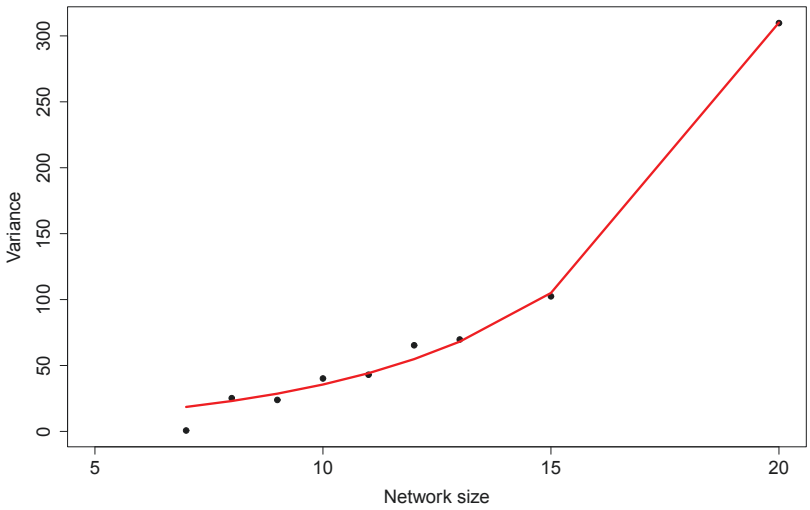


Figure 5.3 Variances of the empirical unimodal distributions over size. Nonlinear regression see Table 5.1.

To obtain a comparable estimate in absolute numbers the distribution is multiplied by a scaling factor, such that the maximum reaches the empirically determined maximum average number of code pairs. This scaling factor grows exponentially with increasing network size (Figure 5.4) in accordance with the maximum. The scaling factor and the variance are both modelled by

$$a \cdot b^s,$$

where a and b are estimated from the data using the `nsl` method in R.
The scaling factor is determined an iterative procedure until the maximum (determined by the R function `optimize`, package `stats`) of the distribution (calculated by the R functions `dnorm`,`dlnorm` and `dgamma`, package `stats`) reaches the maximum value in the data (with a precision of 10^{-2}) (see Algorithm A.12 on page 115).
The general form of the overall model is given by

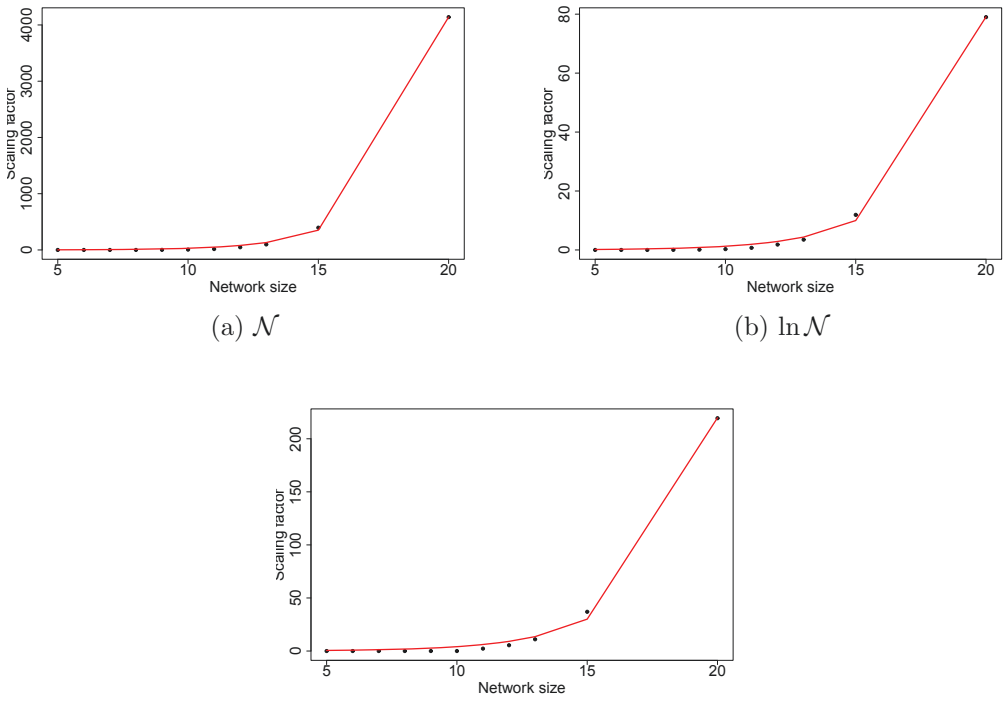
$$\widehat{SC(s,d)}_0^{\mathcal{D}} = \widehat{f_{\mathcal{D}}(s)} \cdot \mathcal{D}\left(d; \theta_1, \theta_2\right), \tag{5.9}$$

where $\mathcal{D}$ denotes one of the candidate distributions and θ_1 and θ_2 the two parameters as calculated for the distributions (see above). For an arbitrary combination of size and density Eq. 5.9 gives the null model estimate for the semantic capacity applying the parameters summarised in Table 5.1.

Goodness of fit. I estimated the goodness of fit on the data by calculation of the euclidean distance $\Delta(data, \mathcal{D})$ between the data and the model prediction for each network size s given by

$$\Delta_s(data, \mathcal{D}) = \sqrt{\sum_r (x_s^r - \widehat{SC(s,d)}_0^{\mathcal{D}})^2},$$

where x_s^r denotes the average number of BMCs identified in random networks of size s and density r.



(a) $\mathcal{N}$ (b) $\ln\mathcal{N}$

(c) Γ

Figure 5.4 Scaling factors of the used distributions over size. Parameters of the superimposed non-linear fit see Table 5.1.

Figure 5.5 shows the results of the analysis. It can be observed that the gamma distribution has the lowest mean distance over the complete dataset ($\Delta_s(\bar{data}, \Gamma) \approx 1$), while the normal distribution is not well suited ($\Delta_s(\bar{data}, \mathcal{N}) \approx 5$) to model the data. The log-normal model has a mean euclidean distance between the normal and the gamma model ($\Delta_s(\bar{data}, \log\mathcal{N}) \approx 2.5$), but also does not fit the data well. The gamma distribution seems to fit well for most of the sampled network sizes, such that $\mathcal{X} \sim \Gamma$ can be assumed. $\widehat{SC(s,d)}_0^\Gamma$ is the corresponding statistical model describing the distribution of code pairs in random networks. The model allows to some extend a prediction of the number of code pairs for random reaction networks with network sizes covered by the used dataset (Figure 5.6). Nevertheless, the model is not perfectly fitted and a prediction over- (for smaller networks) or underestimates (for larger networks) the optimal

Table 5.1 Summary of the statistical models.

Model	b	p-val	a	p-val	R^2
$\widehat{\mu(s)} = b + a \cdot s$	-8.80	$p < 0.001$	2.62	$p < 0.001$	0.87
	b	p-val	a	p-val	residual std. err.
$\widehat{\sigma^2(s)} = a \cdot b^s$	1.24	$p < 0.001$	4.08	$p < 0.001$	8.36 (df=7)
$\widehat{f_{\mathcal{N}}(s)} = a \cdot b^s$	1.64	$p < 0.001$	0.22	$p < 0.005$	26.23 (df=9)
$\widehat{f_{\ln\mathcal{N}}(s)} = a \cdot b^s$	1.51	$p < 0.001$	0.02	$p < 0.01$	1.00 (df=9)
$\widehat{f_{\Gamma}(s)} = a \cdot b^s$	1.49	$p < 0.001$	0.08	$p < 0.05$	3.55 (df=9)

interval and the maximum number of code pairs. Figure 5.7 shows the gamma-model's behaviour for combinations of sizes 1 to 40 and densities 1 to 200.

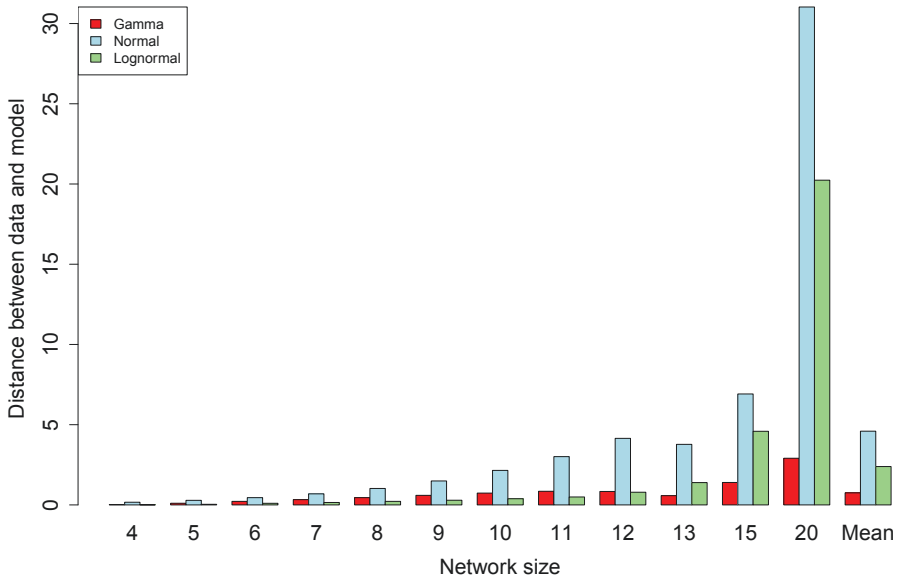


Figure 5.5 Goodness of fit of the three candidate distributions. Measured by the euclidean distance $\Delta_s(data, model)$ for each network size s. The right-most group of bars shows the mean value over all sizes. The normal distribution shows the worst fit over all sizes, while the gamma model has the best fit over only four sizes. The overall good fit of the gamma model is due to its good fit at large networks compared to the other models.

The model behaviour for network sizes larger than 20 shows that the model looses its unimodal form (approx. at size 34) and the maximum does not follow the linear trend any more (approx. at size 25). Thus, the model can not be applied for the prediction of network sizes larger then 25, which is a critical value here. The observed behaviour is typical for the gamma distribution for certain combinations of the parameters scale and shape.
To summarise the analysis: Random reaction networks can be used as a null-model for molecular codes. If a biological system would be under no further constraints, but completely determined by random processes, the system's ability to realise molecular codes would be completely described by the null-model. The gamma distribution showed to be a good statistical model for smaller network sizes, but is not a good prediction model for networks larger than 25.

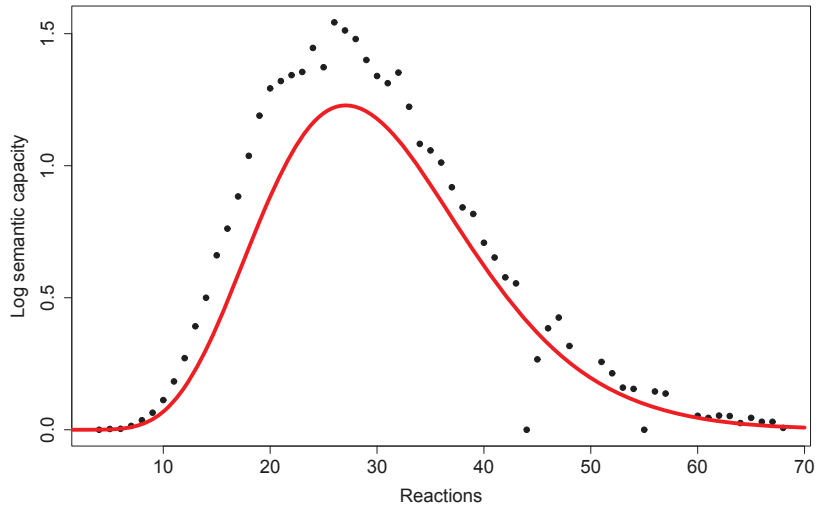


Figure 5.6 Data and model (gamma) overlay. Here shown for random networks of size 15. The deviance between model and data corresponds with the goodness of fit (cf. 5.5).

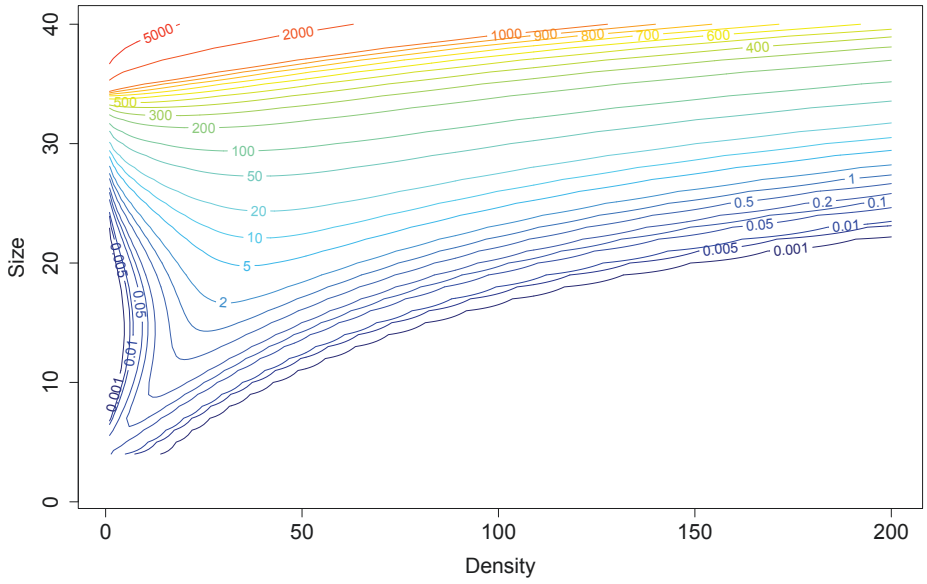


Figure 5.7 Prediction of the statistical null model. Prediction of the log semantic capacity of the statistical model $\widehat{SC(s,d)_0}^{\Gamma}$ for sizes between 1 and 40 and densities between 1 and 200. The curve shows a unimodal behaviour (for constant size over density) until size 25 and the switches to an exponential form.

5.2 Combustion chemistries

The code definition can be applied to any kind of system. Here I will analyse network models of several combustion chemistries. A combustion chemistry describes all chemical reactions happening during the burning of a certain chemical species, e.g. ethanol. The network models I will analyse here are from different sources (cf. 5.2) and are considered to contain all relevant reactions. The prerequisites necessary for a code based analysis are fulfilled for combustion chemistries, because all chemical species that can occur are included and also all possible reactions that can happen under the given physical conditions of combustion, e.g. temperature, are included. Most of the reactions are reversible, such that the network models contain two reactions for the two directions (compare also the networks in Appendix E 2).
The reaction network models cover different sizes (10 - 79 molecular species) and densities (38 - 752 reactions). The code based analysis shows that none of these chemistries is able to realise molecular codes. The statistical null model cannot be applied here to compare the results with the random expectation, since the network sizes are out of the prediction range of the statistical null model. To allow a comparison with a null model I generated random networks of the same size and density and computed the mean number of BMCs, for each combustion chemistry, respectively.
For the hydrogen chemistry, in general, the lack of code pairs can be explained by the small number of closed sets compared to the number of paths, such that the molecular species are "too connected" and the network is less structured. In the null model also no molecular codes can be identified. The estimated number of closed sets and paths, although differing from the original chemistry, are also marking that the respective random networks are not in the optimal interval.
In the methane combustion chemistry there exist far more paths than closed sets, such that the network is to some extend "unstructured". The according null model networks also contain a high number of paths, but also a higher number of closed sets. The algorithmic analysis shows that some of the generated null model networks can realise BMCs, with an average logarithmic semantic capacity of 1.04. Assuming that the maximum number of codes of the null model increases exponentially (cf. Section 5.1) a semantic capacity of 1 can be considered to be very low.

Table 5.2 Overview of the analysed combustion chemistries.

Network	Reference	$\lvert\mathcal{M}\rvert$	$\lvert\mathcal{R}\rvert$	#paths	#closed sets	Sc_{log}
Dimethyl ether	[71]	79	708	$> 10^6$	8	0
Ethanol	[72]	57	752	$> 10^6$	5136	0
Hydrogen	[73]	10	38	$> 10^4$	16	0
Methane	[74]	37	340	$> 10^6$	4136	0

5.3 The artificial chemistry NTOP

Recall that with increasing density random networks have a vanishing semantic capacity. In the following I will show that even a dense network can have a relatively high semantic capacity. For this purpose I analysed an artificial chemistry with 16-species introduced by Banzhaf [75] called NTOP. For each species there is a 4-bit binary representation and the reaction rules are derived with respect to this representation, which is referred to as a structure-to-function mapping (see [75] for details and Appendix E 3 for the network model).
The algorithmic analysis results in six code pairs (Figure 5.8) . Two properties of molecular codes that are of general importance also for biological molecular codes can be observed here. (1) A meaning can take the role of a sign in another code (MSCL-type linkage), and (2) molecular species can function as signs (or meanings) in different codes, i.e. they keep their role in different contexts.

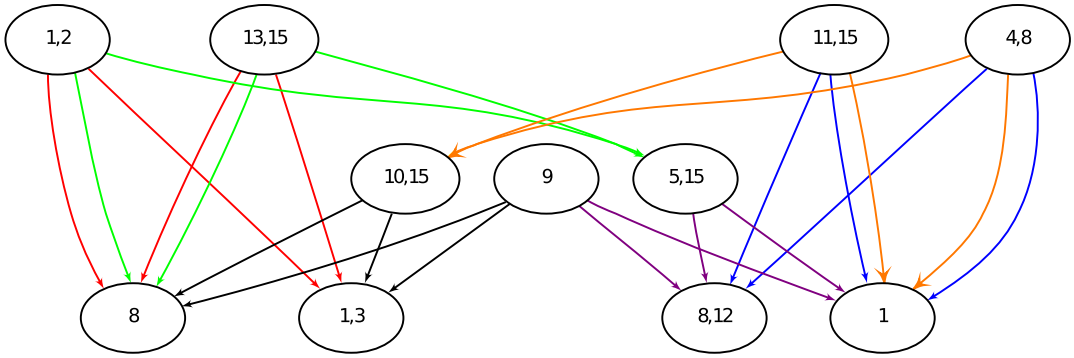


Figure 5.8 Codes in the artificial chemistry NTOP. The six codes have been coloured differently. Contexts have been omitted.

To test the robustness of the network's semantic capacity, 1, 2, 5, 10, 15, 200, and 1000 reaction rules have been replaced randomly (100 replicates), respectively. In a randomly chosen reaction rule only the molecular species are replaced, while the number of reactants and products is kept the same. In the whole network the degree distribution stays the same, while the actual connections are changed. Increased randomisation results in a decreased average semantic capacity (Figure 5.9). The general trend towards less code pairs can be explained by referring to the random reaction networks analysis. Random reaction networks with the same number of species and reactions as NTOP show no semantic capacity ($SC_{log} = 0$). The random variation of the NTOP chemistry drives the system towards the mean semantic capacity of random networks. For systems that are under the effect of some kind of random variation, e.g. mutations, similar conclusions can be drawn. So it may be possible that a system that is located in the optimal interval for random code generation could by chance acquire more codes (structurally) it if is under the effect of random variation.

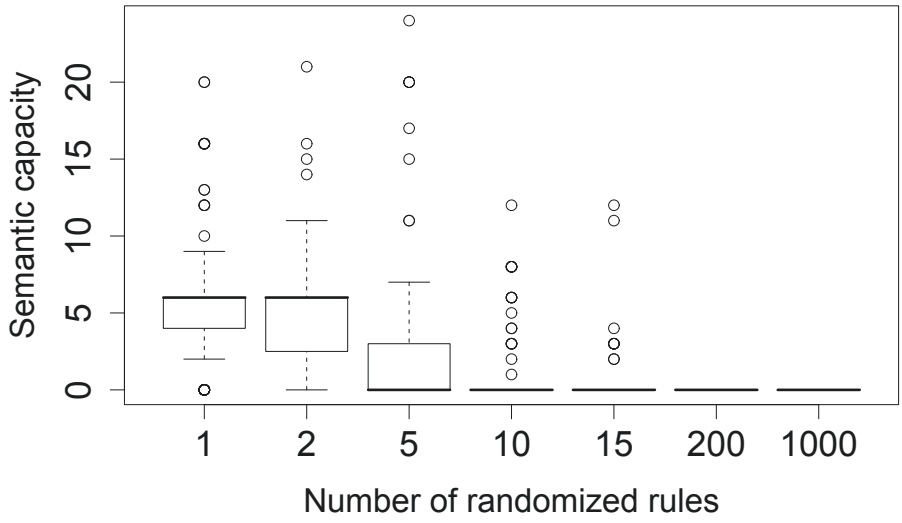


Figure 5.9 Semantic capacity of NTOP under growing randomisation. The randomisation experiment shows that, with growing randomisation, the network's semantic capacity converges towards the null model estimate (= 0 code pairs).

5.4 Photochemistry of Mars

I analysed a model of the photochemistry of planet Mars[76]. The same network has been already analysed in the context of chemical organisation theory [77, 78]. The network can be used to model day and night-side of Mars by adding, or taking out the inflow reaction of light ($\rightarrow h\nu$). As has been demonstrated in [77] this leads to two totally differently structured chemistries, in terms of closed sets and organisations. Thus it may be promising to investigate both network versions also in terms of molecular codes. Both models contain 32 molecular species, i.e. light is also a molecular species in the night side model, and 103 and 104 reactions, respectively.
The day side model is rather easy to compute with the closure based algorithm and does not contain any molecular codes. The night side model shows a totally different picture. The pathway based algorithm with $K = 20$ results in 26 molecular codes. A further analysis of the resulting codes showed that all used either $h\nu$, e[2] in the domain or part of the molecular context. Since light should not be present during night these codes only can work if another light source, perhaps locally, would be present. Without light on the night side these code are not feasible. To check whether the network keeps its capacity to realise molecular codes during night I constructed a second reaction network model of the night side by completely deleting all reactions using light as reactant (Table 5.3) and repeated the analysis. The modified network contains 31 molecular species and 76 reactions, but no codes any more.
The example of the Marsian photochemistry shows that a validation of the codes found, either by structural, or by dynamical arguments is very important for the code based analysis.

[2]Free electrons e can only be produced using light in the model.

Table 5.3 Light consuming reactions in the Mars photochemistry.

Reaction
$1\ O_2\ 1\ h\nu \rightarrow 2\ O$
$1\ O_2\ 1\ h\nu \rightarrow 1\ O\ 1\ O(^1D)$
$1\ O_3\ 1\ h\nu \rightarrow 1\ O_2\ 1\ O$
$1\ O_3\ 1\ h\nu \rightarrow 1\ O_2\ 1\ O(^1D)$
$1\ O_3\ 1\ h\nu \rightarrow 3\ O$
$1\ H_2\ 1\ h\nu \rightarrow 2\ H$
$1\ OH\ 1\ h\nu \rightarrow 1\ O\ 1\ H$
$1\ HO_2\ 1\ h\nu \rightarrow 1\ OH\ 1\ O$
$1\ H_2O\ 1\ h\nu \rightarrow 1\ H\ 1\ OH$
$1\ H_2O\ 1\ h\nu \rightarrow 1\ H_2\ 1\ O(^D)$
$1\ H_2O\ 1\ h\nu \rightarrow 2\ H\ 1\ O$
$1\ H_2O_2\ 1\ h\nu \rightarrow 2\ OH$
$1\ CO_2\ 1\ h\nu \rightarrow 1\ CO\ 1\ O$
$1\ CO_2\ 1\ h\nu \rightarrow 1\ CO\ 1\ O(^1D)$
$1\ NO\ 1\ h\nu \rightarrow 1\ N\ 1\ O$
$1\ NO_2\ 1\ h\nu \rightarrow 1\ NO\ 1\ O$
$1\ NO_3\ 1\ h\nu \rightarrow 1\ NO_2\ 1\ O$
$1\ NO_3\ 1\ h\nu \rightarrow 1\ NO\ 1\ O_2$
$1\ N_2O\ 1\ h\nu \rightarrow 1\ N_2\ 1\ O(^1D)$
$1\ N_2O_5\ 1\ h\nu \rightarrow 1\ NO_2\ 1\ NO_3$
$1\ HNO_2\ 1\ h\nu \rightarrow 1\ OH\ 1\ NO$
$1\ HNO_3\ 1\ h\nu \rightarrow 1\ NO_2\ 1\ OH$
$1\ HO_2NO_2\ 1\ h\nu \rightarrow 1\ HO_2\ 1\ NO_2$
$1\ O\ 1\ h\nu \rightarrow 1\ O^+\ 1\ e$
$1\ O_2\ 1\ h\nu \rightarrow 1\ O_2^+\ 1\ e$
$1\ CO_2\ 1\ h\nu \rightarrow 1\ CO_2^+\ 1\ e$
$1\ CO_2\ 1\ h\nu \rightarrow 1\ CO\ 1\ O^+\ 1\ e$

For the complete model see Appendix E.

5.5 The genetic code

The genetic code, i.e. the mapping describing the translation from nucleotide triplets to amino acids, was the first biological code described as such [79] and is often used as initial example for molecular codes [16, 23, 80].
To check whether the genetic code is a molecular code (Definition 3.1.8) I will identify contingent molecular mappings in the reaction network describing the translation from codons to amino acids. In recent species mainly one code is realised leading to the notion of the "universal genetic code" [81, 17]. Because of this the reaction network that describes gene translation only contains one of the potential mappings between codons and amino acids, but lacks (all) alternative ones. For the algorithmic code identification such a network model is useless. One approach to overcome this effect is to merge the known genetic codes in one reaction network, such that the merged network contains all known alternatives. The fact that there exist more than one genetic code is known for a long time [82, 83]. The 17 known genetic codes, as listed at NCBI [84], cover nuclear and non-nuclear codes of different genera, e.g. bacterial, archaeal, and plant plastid codes, the vertebrate, invertebrate and yeast mitochondrial codes, and the alternative yeast nuclear code. To merge the known genetic codes I construct a reaction network containing the 64 codons, 20 amino acids, and the specific tRNAs, which are necessary for the translation. For all mappings between DNA triplets and amino acids occurring in the 17 codes I added a reaction of the form $codon + tRNA \rightarrow amino\ acid$.

The obtained reaction network contains 234 molecular species and 85 reactions.
The algorithmic analysis of this network identified 16 binary molecular codes, i.e. a log semantic capacity of $Sc_{\log} = 4.09$. The binary codes can partly be assigned to larger molecular codes. For instance, the codons CTT,CTG,CTA, and CTC can be mapped on leucin (L) and threonin (T) and give rise to six of the found BMCs. A second group involves the mapping between AGG,AGA and glycin (G), serine (S), arginine (R) and the translation stop. This code can also be decomposed into six BMCs. There does exist four more BMCs that involve the codons TCA, TTA, TAG and TAA and the amino acids leucine (L), glutamine (Q) and the stop signal. The data suggests that it is easier for the cell to change the mapping for the stop signal, than for an amino acid. Table 5.5 summarises the identified BMCs. The general existence of alternative mappings in the genetic translation system suggests that the genetic code qualifies as a molecular code. The relatively small semantic capacity of the merge network demonstrates that the genetic code, thus a principally contingent system, is under strong constraints, regarding the assignment between codons and amino acids. This is in-line with studies that propose certain regularities in the code as for example reviewed in [17].
To calculate the system's potential maximum semantic capacity I extended the reaction network model by including all potential mappings between codons and amino acids, even if they have not been observed so far. The model includes all possible tRNA molecules, such that each codon could be read for each amino acid. The number of binary molecular codes can be calculated. The code decomposition lemma (Lemma 3.2.2) states that complete molecular codes can be decomposed into BMCs and that each pair of elements from the domain forms a code pair with each pair of elements of the codomain. There exist $\binom{64}{2}$ pairs of codon triplets and $\binom{20}{2}$ pairs of amino acids. The number of BMCs is

$$SC(gene\ translation) = \binom{64}{2} \cdot \binom{20}{2} = 383,040. \tag{5.10}$$

The logarithmic semantic capacity is ≈ 18.55. The difference to the merge network (which relies completely on observed variation in the code) suggests that cells use only a small fraction of their semantic capacity and that the code is under evolutionary constraints. In the literature there exists a set of hypotheses, characterising such constraints, on the evolution of the genetic code, e.g. the coevolution theory as discussed in [85].
In the two models above the tRNAs are the adapters and carry the combinatorial complexity of the system. In the following I will analyse a more realistic model of the gene translation machinery by including the loading step of the tRNA. The refined network model $N_{GC} = \langle \mathcal{M}_{GC}, \mathcal{R}_{GC} \rangle$ contains all possible mappings between the 64 codons and 20 amino acids as described above. Additionally, I model the loading step of the tRNAs by inserting the respective amino acyl tRNA synthetases (aaRS) (Figure 5.10). The reaction network N_{GC} describes the core molecular mechanism realising the standard genetic code and all alternative codes. The set of molecular species $\mathcal{M}_{GC}$ contains all DNA strings of length three (Table 5.4, Eq. 2), representing the codons, the twenty proteinogenic amino acids in their free form (Table 5.4, Eq. 3), the twenty amino acids bound in a protein (Table 5.4, Eq. 4), all possible tRNAs in their unloaded (Table 5.4, Eq. 5) and loaded form (Table 5.4, Eq. 6) and all possible aaRS (Table 5.4, Eq. 7), such that the system is able to load all amino acids to all tRNAs.
The set $\mathcal{R}_{GC}$ contains all reactions loading the amino acids onto the tRNAs (Table 5.4, Eq. 8) and all reactions inserting an amino acid in the peptide sequence (Table

Table 5.4 Definition of the gene translation chemistry with synthetases.

Eq.	Definition	Description
1	$\mathcal{M}_{GC} = Codons \cup AA^{free} \cup AA^{prot} \cup aaRS \cup tRNA^{free} \cup tRNA^{loaded}$	Definition of the molecular species in the network
2	$Codons = \{\text{A,C,G,T}\}^3 = \{\text{AAA}, \text{AAC}, \ldots, \text{TTT}\}$	Set representing the 64 codons of the genetic code
3	$AA^{free} = \{\text{Ala}^{\text{free}}, \text{Arg}^{\text{free}}, \text{Asp}^{\text{free}}, \ldots, \text{Try}^{\text{free}}\}$	Amino acids that are not used in a protein
4	$AA^{prot} = \{\text{Ala}^{\text{prot}}, \text{Arg}^{\text{prot}}, \text{Asp}^{\text{prot}}, \ldots, \text{Try}^{\text{prot}}\}$	Amino acids that have been used in a protein during gene translation
5	$tRNA^{free} = \{\text{tRNA}_n \mid n \in Codons\}$	Unloaded tRNAs specific for codon n
6	$tRNA^{loaded} = \{\text{tRNA}_{n,a} \mid n \in Codons, a \in AA_{free}\}$	tRNAs specific for codon n that have been loaded with amino acid a
7	$aaRS = \{\text{Syn}_{n,a} \mid n \in Codons, a \in AA_{free}\}$	Amino acyl-tRNA-synthetases that are specific for amino acid a and codon n
8	$\mathcal{R}_{GC} = \{\text{tRNA}_n + a + \text{Syn}_{n,a} \rightarrow tRNA_{a,n} + Syn_{n,a} \mid n \in Codons, a \in AA^{free}\} \cup$	Loading of the tRNA by suitable synthetase
9	$\{n + \text{tRNA}_{a,n} \rightarrow n + \text{tRNA}_n + a \mid n \in Codons, a \in AA^{prot}\}$	Translation step, i.e., the incorporation of an amino acid into a growing protein

5.4, Eq. 9). Figure 5.10A displays a subnetwork with two codons (GGA, AGU), two amino acids (Gly, Ser) and the respective other elements of the network (tRNA and synthetases). Analysing this subnetwork allows to assess the whole network's semantic capacity. Table 5.6 shows the four molecular code pairs contained in the subsystem, the respective molecular contexts are listed in Table 5.7. The core code analysis of these networks reveals that each single code is only a core code of itself (reflexivity), but never a core code of any other code. In other words, the four codes are not generated by one of the other codes, but stand on their own. The identified code pairs (Table 5.6) show that not only codons can be signs, but also the unloaded tRNAs can function as signs. These additional signs increase the number of code pairs in a combinatoric manner. The "new" codes differ structurally in their molecular context. While, classically, the codons are mapped to the set of amino acids using the loaded tRNAs as context, the new signs, i.e. unloaded tRNAs, are mapped to the set of amino acids by using a molecular context that consists of the free amino acid loaded to the free tRNA, the synthetase performing the loading step, and the codon that needs to be recognised by the tRNA. The number of code pairs in this system can be calculated by

$$CP_{GC} = \left[\binom{n_s}{2} - \frac{n_s}{2} \right] \cdot \binom{n_m}{2}, \tag{5.11}$$

with n_s as number of signs and n_m as number of meanings (amino acids). For the full gene translation system the number of signs is $n_s = c + t$, with c as number of codons and t as number of unloaded tRNAs. Because there is always one pair of one tRNA and one codon belonging together that can not be combined as signs in a BMC, we have to subtract the number of such pairs $n_s/2$ from the amount of all combinations.
The analysis of the whole network (N_{GC}), describing all potential genetic codes with 64 codons and 20 amino acids, results in $1,532,160$ binary code pairs, i.e. $Sc_{\log}(N_{GC}) \approx 20.55$. This is a different result than for the less detailed model, as calculated by Eq. (5.10). The extension of the model by aaRS, unloaded tRNAs, and unloaded amino acids increases the semantic capacity.

The question if and how a tRNA based code could be employed by the cell is open, but the potential existence of such a code is nevertheless an interesting result.

Table 5.5 Molecular codes in the known genetic codes.

sign (codons)	meanings (amino acids)	#BMC	References
CTT, CTG, CTA, CTC	L, T	6	[82, 86]
AGG, AGA	G,S,R, Stop	6	[87, 88, 89, 90, 91, 92, 82, 93, 94, 95, 96, 97, 98]
AGG, TCA	S, Stop	1	[89, 90, 82, 93, 95, 99]
AGA, TCA	S, Stop	1	[89, 90, 82, 93, 95, 99]
TTA, TAG	L, Stop	1	[82, 100, 101, 99, 84]
TAA, TAG	Q, Stop	1	[82, 102, 103, 104, 105]

Here the 16 found BMCs in the merge of the 17 known genetic codes are summarised. If applicable BMCs are grouped. References: Articles reporting the respective alternatives in the genetic code that are part of a BMC in this analysis.

Table 5.6 Code pairs in the gene translation model.

#	Signs	Meanings
1	$\{\mathrm{GGA}, \mathrm{AGU}\}$	$\{\mathrm{Gly}^{\mathrm{prot}}, \mathrm{Ser}^{\mathrm{prot}}\}$
2	$\{\mathrm{GGA}, \mathrm{tRNA}_{\mathrm{AGU}}\}$	$\{\mathrm{Gly}^{\mathrm{prot}}, \mathrm{Ser}^{\mathrm{prot}}\}$
3	$\{\mathrm{AGU}, \mathrm{tRNA}_{\mathrm{GGA}}\}$	$\{\mathrm{Gly}^{\mathrm{prot}}, \mathrm{Ser}^{\mathrm{prot}}\}$
4	$\{\mathrm{tRNA}_{\mathrm{GGA}}, \mathrm{tRNA}_{\mathrm{AGU}}\}$	$\{\mathrm{Gly}^{\mathrm{prot}}, \mathrm{Ser}^{\mathrm{prot}}\}$

Code pairs realised by the subsystem of the gene translation network with synthetases shown in Figure 5.10.

Table 5.7 Molecular contexts of the codes in the gene translation model.

#	Molecular context	alternative molecular context
1	$\{\mathrm{tRNA}_{\mathrm{GGA,Gly}}, \mathrm{tRNA}_{\mathrm{AGU,Ser}}\}$	$\{\mathrm{tRNA}_{\mathrm{AGU,Gly}}, \mathrm{tRNA}_{\mathrm{GGA,Ser}}\}$
2	$\{\mathrm{AGU}, \mathrm{Ser}^{\mathrm{free}}, \mathrm{Syn}_{\mathrm{AGU,Ser}}, \mathrm{tRNA}_{\mathrm{GGA,Gly}}\}$	$\{\mathrm{AGU}, \mathrm{Gly}^{\mathrm{free}}, \mathrm{Syn}_{\mathrm{AGU,Gly}}, \mathrm{tRNA}_{\mathrm{GGA,Ser}}\}$
3	$\{\mathrm{GGA}, \mathrm{Ser}^{\mathrm{free}}, \mathrm{Syn}_{\mathrm{GGA,Ser}}, \mathrm{tRNA}_{\mathrm{AGU,Gly}}\}$	$\{\mathrm{GGA}, \mathrm{Gly}^{\mathrm{free}}, \mathrm{Syn}_{\mathrm{GGA,Gly}}, \mathrm{tRNA}_{\mathrm{AGU,Ser}}\}$
4	$\{\mathrm{GGA}, \mathrm{AGU}, \mathrm{Gly}^{\mathrm{free}}, \mathrm{Ser}^{\mathrm{free}}, \mathrm{Syn}_{\mathrm{GGA,Gly}}, \mathrm{Syn}_{\mathrm{AGU,Ser}}\}$	$\{\mathrm{GGA}, \mathrm{AGU}, \mathrm{Gly}^{\mathrm{free}}, \mathrm{Ser}^{\mathrm{free}}, \mathrm{Syn}_{\mathrm{GGA,Ser}}, \mathrm{Syn}_{\mathrm{AGU,Gly}}\}$

Molecular contexts of the code pairs shown in Table 5.6.

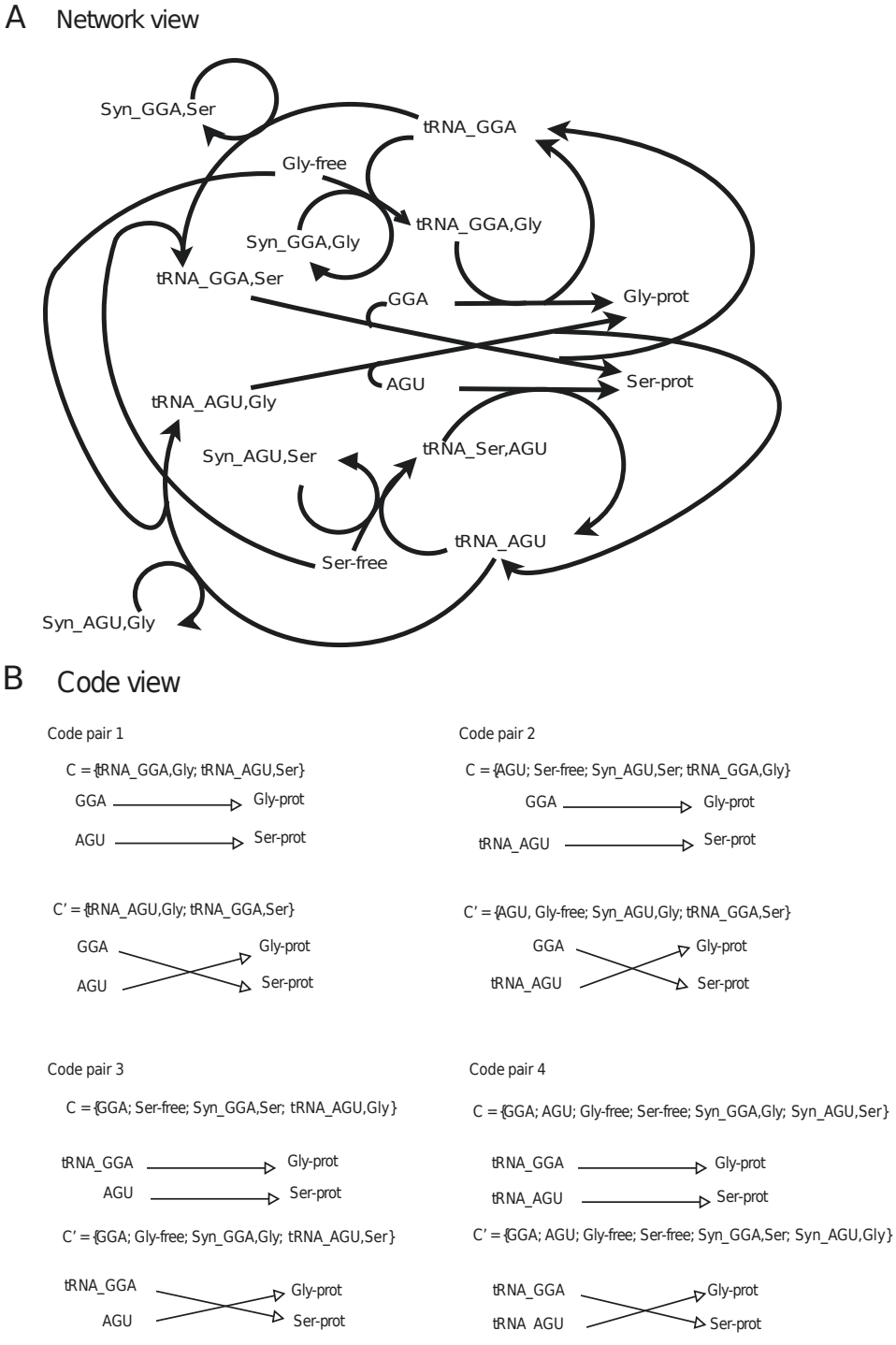


Figure 5.10 Subnetwork of the full gene translation network model with synthetases ($N_{GC'}$) and the realised molecular codes. The network (panel A) shows a subnetwork of the gene translation network model containing the translation, and loading reactions for two selected codons (GGA, AGU) and amino acids (Gly, Ser). The semantic analysis shows that four code pairs can be implemented by this network (panel B).

5.6 Gene regulatory networks

Biological background Cell's maintain a complex regulatory system to orchestrate the expression of their genes. Different information about the external environment and internal states are integrated to regulate the expression of proteins and enzymes. Regulation of gene expression is implemented differently in eukaryotes and prokaryotes, but share a common mechanism: proteins (transcription factors) need to bind the DNA to either activate or repress gene translation. In eukaryotes this process is much more complex, because also protein complexes are formed for this purpose. The gene regulatory system of a cell is also a highly semantic system, because it carries and uses the information about the environment and internal (metabolic) states. This can be seen by analysing a gene regulatory network using the proposed algorithms.

A model of gene regulation To apply the code identifying algorithms at first a network model needs to be developed. In general, gene regulatory networks (GRN) are graphs representing the regulation of the expression of certain genes by the expression of other genes. A node in a GRN stands for a complex process including the gene, the promoter and binding region of that gene, the binding of the transcription factor (TF) plus cofactors and the production of a product by the recruitment of the gene expression machinery. A cell's GRN is also a highly semantic system based in molecular codes. For the analysis a GRN is modelled as reaction network $\mathcal{N}_{GRN} = \langle \mathcal{M}_{GRN}, \mathcal{R}_{GRN} \rangle$ by explicitly inserting the relevant components (Fig 5.11). The resulting network is not a generic model to describe all possible gene regulatory networks, but a model that covers the main properties of regulation important for this study. $\mathcal{M}_{GRN}$ contains n transcription factors TF_i, m products P_j, and genes G_{ij}. Each gene G_{ij} represents a combination of a promoter site i and a coding region j, where the promoter site i is specific to TF_i and the coding region j produces P_j. For the model I assume that there exist as many promoter sites and coding regions as transcription factors and products, respectively, such that each promoter-gene combination is possible. In summary

$$\begin{aligned}\mathcal{M}_{GRN} = \{&TF_1, TF_2, \ldots, TF_i, \ldots, TF_n, P_1, P_2, \ldots, \\ &P_j, \ldots, P_m, G_{11}, G_{12}, \ldots, G_{ij}, \ldots, G_{nm}\}.\end{aligned}$$

The differences of eukaryotic and prokaryotic gene regulation, here, plays only a minor role (and is not modelled) since only the general mechanism of transcription factor regulated expression shall be explored in a very basic approach.
For the abstract model I will assume that a transcription factor binds only one promoter and that a promoter is bound by only one transcription factor. The assumption, that one TF bind specifically only one promoter, and vice versa, is a broad simplification of the real biological system. Nevertheless, for the proof of principle presented here it is a reasonable one. The model could be made more complex (see below), but here it is sufficient to describe the simpler model. The expression of a gene i, j then is given by

$$\begin{aligned}\mathcal{R}_{GRN} = \{TF_i + G_{ij} \rightarrow TF_i + G_{ij} + P_j\}, i = 1, 2, \ldots, n, \\ j = 1, 2, \ldots, m.\end{aligned}$$

Semantic analysis The semantic analysis shows that the reaction network can implement molecular codes, but only in one way, i.e. with the transcription factors as signs

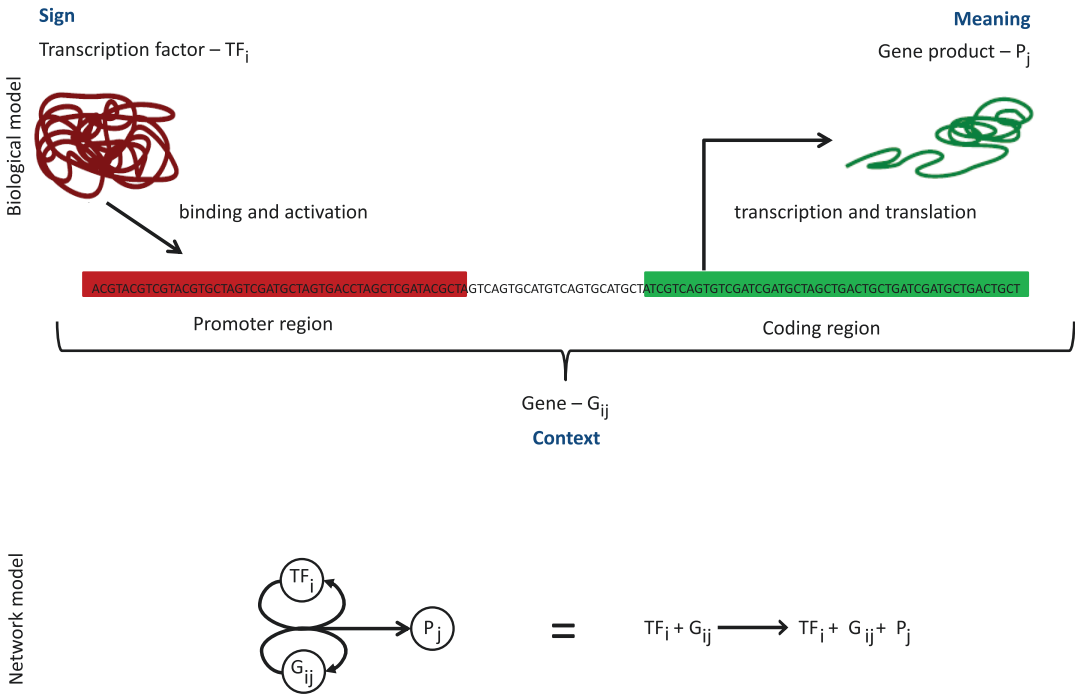


Figure 5.11 Construction of a gene regulatory network model. Biological model of the expression of a gene, and the reaction network formulation of the same process (below). Blue text in panel A indicates the semantic interpretation according to the code based analysis, i.e. the transcription factors are the signs, the products are the meanings, and the DNA is the molecular context.

and the set of products as meanings. The set of genes, i.e. the combination of promoter and coding region, forms the molecular context. So the mapping between transcription factor and gene product can be altered by the exchange of a promoter region of a gene (or vice versa). Such promoter exchanges are also a common tool in molecular biology to allow for the external control of gene expression [106], e.g. to discover the function of silenced gene clusters [107].
Interestingly, in contrast to the model of the gene translation chemistry described above, the DNA is not the sign, but functions as the molecular context. This "role change" suggests an interdependence between different codes. Here the "gene regulatory code" regulates the execution of the "gene translation code", as the former one controls the usage of the latter's signs.
Please note that the reaction network model can easily be made more complex by modelling transcription factors as protein complexes and including the respective assembly processes, by modelling different types of transcription factors (activators, repressors, enhancers), or the introduction of several DNA binding sites in the regulatory region to allow a combinatoric regulation by several transcription factors.
The core code analysis of the GRN network model yields the same result as for the gene translation system, i.e. since the model is quite abstract no nested codes (beside reflexivity) can be identified here.

Linking gene regulation with gene translation I extended the model by linking the genetic code (and all its alternatives) and the gene regulatory code to see how the

semantic capacity changes.
A subnetwork of the model consists of two transcription factors TF1 and TF2, two binding domains (promoters) P1 and P2, two coding regions of the genes, which are modelled explicitly as strings ABA and BAB. The "nucleic acids" A and B can be translated to two amino acids L and K. As in the model above the two promoters are allowed to be freely combined with the two coding regions resulting in four genes. Resulting in four possible protein products defined by the tRNAs available, LLL, KKK, LKL, and KLK. The resulting reaction network contains 14 molecular species and 16 reaction rules (see Appendix E 5 for the reaction network).
This reaction network contains 13 binary molecular codes (Table 5.8). A closer look to the resulting codes shows that molecular species from both subsystems (GRN, GC) can be used as signs, but only the final gene products can be meanings in these codes. While the molecular species from the GRN part can be combined as signs in one code (Table 5.8, codes 2-5), tRNAs are only combined with tRNAs as signs. In the molecular context all molecular species occur (except of the meanings).
Codes 7 and 8 show that it is possible to implement a code based on one incoming signal (compare [80]). In both codes the signs contain the same promoter region, such that the alternative mappings can only be realised by a change in the genetic code, i.e. the selection of the specific tRNAs in the context.
It is only possible to generate contingent mappings to the non-degenerated case, i.e. when A and B are encoded to different amino acids. The degenerated protein LLL, KKK are never used as meanings.
The network combines several biochemical reactions and thus is only a rough model of the underlying processes. I extended the model by introducing the transcribed gene as intermediate product. By decoupling both processes the number of reactions reduces to 10, while the number of molecular species grows by the two transcripts ABA and BAB (for the network see Appendix E 5). This slightly different model now contains 27 BMCs (Table 5.9). The difference in semantic capacity demonstrates that a code based analysis also is dependent on the level of detail of a given model. Structurally the codes from the simple and the extended model do not differ. The new codes are generated by the meaning-sign-linkage (cp Section 3.4.3), because the transcripts now can be used as signs and meanings in the new codes.

Table 5.8 Codes identified in the combined GC-GRN network.

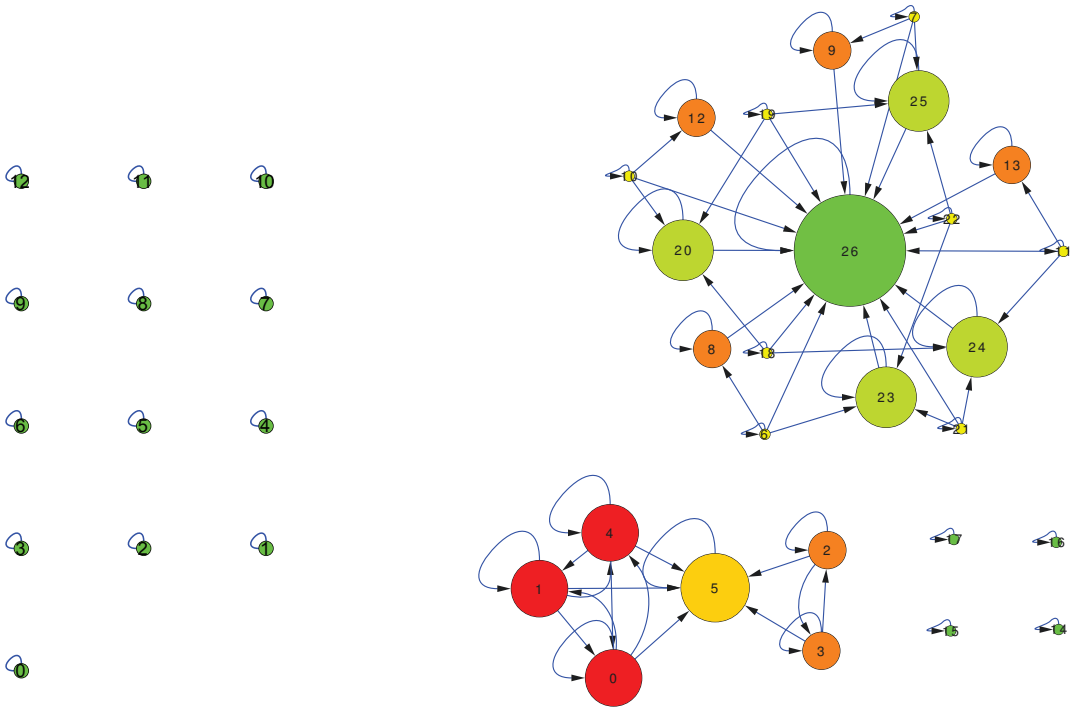
	Domain		Codomain		Molecular contexts	
1	TF1	TF2	LKL	KLK	P1BAB, P2ABA, tRNA_A_K, tRNA_B_L	P1BAB, P2ABA, tRNA_A_K, tRNA_B_L
2	TF1	P2BAB	LKL	KLK	TF2, P1ABA, tRNA_A_L, tRNA_B_K	T2, P1ABA, tRNA_A_K, tRNA_B_L
3	TF1	P2ABA	LKL	KLK	TF2, P1BAB, tRNA_A_K, tRNA_B_L	TF2, P1BAB, tRNA_A_L, tRNA_B_K
4	TF2	P1ABA	LKL	KLK	TF1, P2BAB, tRNA_A_K, tRNA_B_L	TF1, P2BAB, tRNA_A_L, tRNA_B_K
5	TF2	P1BAB	LKL	KLK	TF1, P2ABA, tRNA_A_L, tRNA_B_K	TF1, P2ABA, tRNA_A_K, tRNA_B_L
6	P1ABA	P2BAB	LKL	KLK	TF1, TF2, tRNA_A_L, tRNA_B_K	TF1, TF2, tRNA_A_K, tRNA_B_L
7	P1ABA	P1BAB	LKL	KLK	TF1, tRNA_A_L, tRNA_B_K	TF1, tRNA_A_K, tRNA_B_L
8	P2BAB	P2ABA	LKL	KLK	TF2, tRNA_A_K, tRNA_B_L	TF2, tRNA_A_L, tRNA_B_K
9	P2ABA	P1BAB	LKL	KLK	TF1, TF2, tRNA_A_L, tRNA_B_K	TF1, TF2, tRNA_A_K, tRNA_B_L
10	tRNA_A_L	tRNA_A_K	LKL	KLK	TF1, P1ABA, tRNA_B_K, tRNA_B_L	TF1, P1BAB, tRNA_B_K, tRNA_B_L
11	tRNA_A_L	tRNA_B_L	LKL	KLK	TF1, P1ABA, tRNA_B_K, tRNA_A_K	TF1, P1BAB, tRNA_B_K, tRNA_A_K
12	tRNA_B_K	tRNA_A_K	LKL	KLK	TF1, TF2, P1ABA, , P2ABA, tRNA_A_L, tRNA_B_L	TF1, P1BAB, tRNA_A_L, tRNA_B_L
13	tRNA_B_K	tRNA_B_L	LKL	KLK	TF1, TF2, P1ABA, , P2ABA, tRNA_A_L, tRNA_A_K	TF1, P1BAB, tRNA_A_L, tRNA_A_K

A and B denote the two codons, while L and K denote the two amino acids. P1 and P2 are the two promoter sites specific for TF1 and TF2.

Table 5.9 Codes identified in the extended GC-GRN network.

	Domain		Codomain		Molecular contexts	
0	TF1	TF2	LKL	KLK	P1BAB, P2ABA, tRNA_A_K, tRNA_B_L	P1BAB, P2ABA, tRNA_A_L, tRNA_B_K
1	TF1	TF2	LKL	ABA	P1BAB, P2ABA, tRNA_A_K, tRNA_B_L	P1ABA, P2BAB, tRNA_A_K, tRNA_B_L
2	TF1	TF2	LKL	BAB	P1ABA, P2BAB, tRNA_A_L, tRNA_B_K	P1BAB, P2ABA, tRNA_A_L, tRNA_B_K
3	TF1	TF2	KLK	ABA	P1BAB, P2ABA, tRNA_A_L, tRNA_B_K	P1ABA, P2BAB, tRNA_A_L, tRNA_B_K
4	TF1	TF2	KLK	BAB	P1ABA, P2BAB, tRNA_A_K, tRNA_B_L	P1BAB, P2ABA, tRNA_A_K, tRNA_B_L
5	TF1	TF2	ABA	BAB	P1ABA, P2BAB	P1BAB, P2ABA
6	TF1	P2ABA	LKL	KLK	TF2, P1BAB, tRNA_A_K, tRNA_B_L	TF2, P1BAB, tRNA_A_L, tRNA_B_K
7	TF1	P2ABA	LKL	KLK	TF2, P1ABA, tRNA_A_L, tRNA_B_K	TF2, P1ABA, tRNA_A_K, tRNA_B_L
8	TF1	ABA	LKL	KLK	P1BAB, tRNA_A_K, tRNA_B_L	P1BAB, tRNA_A_L, tRNA_B_K
9	TF1	BAB	LKL	KLK	P1ABA, tRNA_A_L, tRNA_B_K	P1ABA, tRNA_A_K, tRNA_B_L
10	TF2	P1ABA	LKL	KLK	TF1, P2BAB, tRNA_A_K, tRNA_B_L	TF1, P2BAB, tRNA_A_L, tRNA_B_K
11	TF2	P1BAB	LKL	KLK	TF1, P2ABA, tRNA_A_L, tRNA_B_K	TF1, P2ABA, tRNA_A_K, tRNA_B_L
12	TF2	ABA	LKL	KLK	P2BAB, tRNA_A_K, tRNA_B_L	P2BAB, tRNA_A_L, tRNA_B_K
13	TF2	BAB	LKL	KLK	P2ABA, tRNA_A_L, tRNA_B_K	P2ABA, tRNA_A_K, tRNA_B_L
14	tRNA_A_L	tRNA_A_K	LKL	KLK	tRNA_B_L, tRNA_B_K, ABA	tRNA_B_K, tRNA_B_L, BAB
15	tRNA_A_L	tRNA_B_L	LKL	KLK	tRNA_A_K, tRNA_B_K, ABA	tRNA_A_K, tRNA_B_K, BAB
16	tRNA_A_K	tRNA_B_K	LKL	KLK	tRNA_A_L, tRNA_B_L, ABA	tRNA_A_L, tRNA_B_L, BAB
17	tRNA_B_L	tRNA_B_K	LKL	KLK	tRNA_A_L, tRNA_A_K, ABA	tRNA_A_L, tRNA_A_K, BAB
18	P1ABA	P1BAB	LKL	KLK	TF1, tRNA_A_L, tRNA_B_K	TF1, tRNA_A_K, tRNA_B_L
19	P1ABA	P2BAB	LKL	KLK	TF1, TF2, tRNA_A_L, tRNA_B_K	TF1, TF2,tRNA_A_K, tRNA_B_L
20	P1ABA	BAB	LKL	KLK	TF1, tRNA_A_L, tRNA_B_K	TF1,tRNA_A_K, tRNA_B_L
21	P2ABA	P1BAB	LKL	KLK	TF1, TF2, tRNA_A_L, tRNA_B_K	TF1, TF2, tRNA_A_K, tRNA_B_L
22	P2ABA	P2BAB	LKL	KLK	TF2, tRNA_A_L, tRNA_B_K	TF2, tRNA_A_K, tRNA_B_L
23	P2ABA	BAB	LKL	KLK	TF2, tRNA_A_L, tRNA_B_K	TF2, tRNA_A_K, tRNA_B_L
24	P1BAB	ABA	LKL	KLK	TF1, tRNA_A_K, tRNA_B_L	TF1, tRNA_A_L, tRNA_B_K
25	P2BAB	ABA	LKL	KLK	TF2, tRNA_A_K, tRNA_B_L	TF2, tRNA_A_L, tRNA_B_K
26	ABA	BAB	LKL	KLK	tRNA_A_L, tRNA_B_K	tRNA_A_K, tRNA_B_L

A and B denote the two codons, while L and K denote the two amino acids. P1 and P2 are the two promoter sites specific for TF1 and TF2.



(a) Simple model

(b) Extended model

Figure 5.14 Nested codes in the GC-GRN models. Arrow heads point towards the nested code. Node size correspond to number of incoming edges. Color corresponds to the cluster coefficient of each node. Green - small, to red - large. A - None of the identified codes are nested, i.e. their internal structure may overlap, but all codes are different in some of their components. B - In the extended model a certain nesting structure can be observed. The codes labels 5 and 26 are the "pure" GRN and GC. Codes 14 - 17 use some flexibility on the GC for the alternative mappings and thus stand on their own. The other codes are induced by the GRN or the GC part of the model.

5.7 Protein assembly

The notion of adapters as central concept in Barbieri's organic codes [16] and the compositional semantics as proposed by Gimona [11] suggest that the assembly of protein complexes is a cellular subsystem the cell uses for encoding information. I will here analyse a simple protein assembly process.
At first I will analyse a simple toy model of protein assembly where all complexes are allowed to form, i.e. each protein can interact with each other protein. Starting with 2 proteins A and B the set of molecular species is $\{A, B\}$. After the first assembly step the molecular species $\{A, B, AA, AB, BB\}$ are generated. After the second step $\{A, B, AA, AB, BB, AAA, AAB, ABA, ABB, BAB, BBB, AAAA, AAAB, AABB, ABAA, ABAB, ABBA, ABBB, BBAB, BBBB\}$, and so on. Stopping a the second step induces a reaction network (Appendix E 7) that can be used for the analysis.
The algorithm identifies one binary molecular code mapping the initial molecular species A and B to AAB and ABB either by using the context $\{AB\}$ or alternatively $\{AA, BB\}$ This indicates that protein assembly can generate contingent mappings under the assumption that cells can regulate the molecular contexts of the potential codes. This simple example shows that the sign, or the meanings can also be part of the context in one code. Because in biology different complexes have different functions, even if some constituents of the complexes are similar, such codes are not by default infeasible. As for all algorithmically identified codes, also at protein assembly, dynamics and other criteria have to be taken into account to identify feasible codes.
The analysed network here describes the association of proteins and complexes. By modelling also the dissociation for the two step complexation network results in a slightly larger network containing 20 species and 23 reactions (see Appendix E 7.1). This reaction network does not contain codes any more. Inhowfar, this result is representative for actual protein assembly processes needs to be checked in further studies. Sources of errors, here, may be the small network size and the symmetry of the generated networks. Both factors may lead to the effect that dissociation destroys the semantic capacity.

5.8 Signalling by phosphorylation cascades allows for molecular codes only in a dynamic setting

The most prominent signalling systems rely on reversible phosphorylation of amino acids side-chains for regulation of signalling protein activity. The direct involvement of such systems in signalling suggest that they may be semantic systems. If so, they should be able to realise molecular codes. I have studied phosphorylation cascades, like the mitogen activated kinase regulatory network, as a typical instance of an intra-cellular signalling system. These systems demonstrate the limitation of the static approach. Here, it is necessary not only to distinguish between molecular species, but also between their concentrations. By assigning concentration levels to each species I allow for the dynamic change of these by the system's reactions. Thus, a molecular species' concentration is decreased, if it is used as reactant in a reaction and increased if produced. In the reaction network a species can have an effect on another species' concentration through the reactions in the system.
In general, the activation of a kinase by phosphorylation can generate a molecular mapping between the kinase and its target, but this mapping is not necessarily a molecular

code (Figure 5.15A, page 75). In contrast, a two-step cascade is able to implement a molecular code (Figure 5.15B, page 75).
The simple one-step phosphorylation model (Figure 5.15A, page 75) contains two kinases: an initial kinase (S) and a target kinase (A) which can be phosphorylated by S ($S^P + A \rightarrow A^P$). The dephosphorylation step is modelled as spontaneous reaction $A^P \rightarrow A$. Phosphatases, and the phosphate related molecular species (e.g. ATP, ADP, P) involved in the process are not modelled explicitly, but assume as buffered concentration. In the simple one-step model a molecular mapping between S^P and the two states of kinase A can be identified. If S^P has a low concentration the system is in a state where the unphosphorylated state A has a high concentration and the phosphorylated state A^P has a low concentration. According to the definition of molecular code given above the system should be able to change the mapping, i.e. be contingent, by the application of a different molecular context to realise a code. Here, no alternative mapping between S and A can be realised, such that the system is not able to realise a molecular code.
I will also analyse a different system with two kinases between S^P and A, i.e. a two-step phosphorylation cascade (Figure 5.15B, page 75). S^P now phosphorylates the inserted species, while these have an effect on A. Now the system has the possibility to "choose" between two alternative systems, i.e. the inserted species may be "active" in the unphosphorylated state (B), or in the phosphorylated state (C). There exist several mappings in such a system, e.g. between S^P and B, S and C, and S^P and A. The former two mappings behave like the simple model (see above). The mapping between S and A is a molecular code, because the molecular context of the system can be changed, such that the alternative system behaviour is generated (see Figure 5.15B (right), page 75). The molecular context between S and A is either the set $\{B, B^P\}$, or alternatively $\{C, C^P\}$. I assume two concentration levels denoted by $[.]high$ and $[.]low$ for high and low concentrations, respectively. The following codes can be identified: Under the molecular context $\{B, B^P\}$ the mappings $[S^P]low \rightarrow [A]low$, $[S^P]low \rightarrow [A^P]high$, $[S^P]high \rightarrow [A]high$, and $[S^P]high \rightarrow [A^P]low$.
Under the molecular context $\{C, C^P\}$ the mappings $[S^P]low \rightarrow [A]high$, $[S^P]low \rightarrow [A^P]low$, $[S^P]high \rightarrow [A]low$, and $[S^P]high \rightarrow [A^P]high$. Figure 5.15(C) shows a parameter scan of the system under the two contexts. The dynamic model is based on mass action kinetics given by the following system of ordinary differential equations:

$$\begin{aligned}
\frac{\mathrm{d}([\mathrm{A}])}{\mathrm{d}t} &= -(0.1 \cdot [\mathrm{B}] \cdot [\mathrm{A}]) - (0.1 \cdot [\mathrm{C_P}] \cdot [\mathrm{A}]) + (0.1 \cdot [\mathrm{A_P}]) \\
\frac{\mathrm{d}([\mathrm{A_P}])}{\mathrm{d}t} &= +(0.1 \cdot [\mathrm{B}] \cdot [\mathrm{A}]) + (0.1 \cdot [\mathrm{C_P}] \cdot [\mathrm{A}]) - (0.1 \cdot [\mathrm{A_P}]) \\
\frac{\mathrm{d}([\mathrm{B}])}{\mathrm{d}t} &= -(0.1 \cdot [\mathrm{S_P}] \cdot [\mathrm{B}]) + (0.1 \cdot [\mathrm{B_P}]) \\
\frac{\mathrm{d}([\mathrm{B_P}])}{\mathrm{d}t} &= +(0.1 \cdot [\mathrm{S_P}] \cdot [\mathrm{B}]) - (0.1 \cdot [\mathrm{B_P}]) \\
\frac{\mathrm{d}([\mathrm{C}])}{\mathrm{d}t} &= -(0.1 \cdot [\mathrm{S_P}] \cdot [\mathrm{C}]) + (0.1 \cdot [\mathrm{C_P}]) \\
\frac{\mathrm{d}([\mathrm{C_P}])}{\mathrm{d}t} &= +(0.1 \cdot [\mathrm{S_P}] \cdot [\mathrm{C}]) - (0.1 \cdot [\mathrm{C_P}])
\end{aligned}$$

Applying the context $\{B, B^P\}$ an increase in $[S^P]$ (x-axis) leads to a decrease in the $[A^P]/[A]$-ratio (y-axis). Applying $\{C, C^P\}$ leads to the opposite behaviour.

The extension of the static approach to a dynamic setting needs more strict definitions, such that the here shown properties are only a first step into this direction. For the discussion of potential extensions see Chapter 6 "Towards pragmatics"(pp. 85).

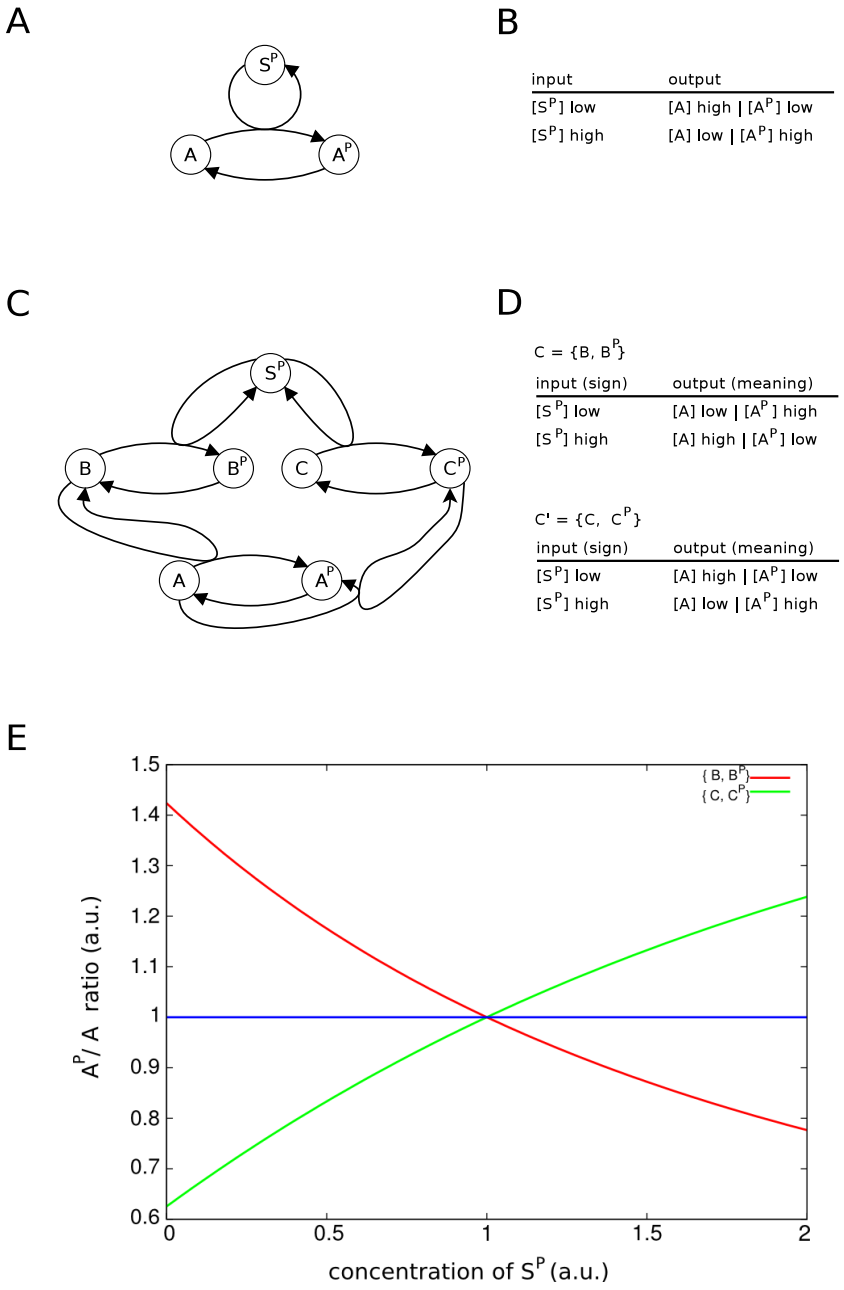


Figure 5.15 Reaction networks describing phosphorylation motifs. Molecular species in these networks represent kinases that may be activated or inactivated by phosphorylation. Activated and non-activated forms of kinase are modelled as different species (e.g. species A and A^P). Panel A: Reaction network of a simple phosphorylation motif, which can realise a molecular mapping (panel B), but not a molecular code. Panel C: more complex reaction network that can realise molecular codes (panel D). Panel D: The two binary molecular codes (one code pair) are realised by either one of the two molecular contexts $\{B, B^P\}$ or $\{C, C^P\}$. In contrast to the other described molecular codes (e.g. the genetic code), here, the code is not only specified by the species, but also by their concentrations. Panel E shows the $[A^P]/[A]$ ratio over $[S^P]$ for the two different contexts. The red line shows the system's behaviour for the context $\{B, B^P\}$, while the green line shows the system's behaviour for the alternative context $\{C, C^P\}$ over varying initial concentrations for S^P. The blue line indicates the (here arbitrary) threshold to separate high and low concentration.

5.9 Analysis of large scale biological networks

I will here present a first code based analysis of two major biological systems, i.e. human signal transduction, and the KEGG metabolic network. The analysis shows that the static definitions presented here need to be coupled with a validation step to identified the feasible codes in the set of all identified potential codes.

5.9.1 Metabolism

For the analysis of metabolism I will use the metabolic network from the KEGG[3] REACTIONS database [108, 109]. The network contains 6777 molecular species and 8182 reactions and covers all biochemical reactions known, i.e. the network is a merge from the different species contained in KEGG. Due to the size of the network the Monte-Carlo subnetwork sampling algorithm is chosen to analyse the network. As parameters I, empirically, determined a subnetwork size of 30, K=6 and a coverage of 10000 as suitable setting with respect to identification power and runtime. The algorithmic analysis identified 37 BMCs (see Table D.1, page 127, for all identified codes). It seems that, from a static point of view, the metabolic network of cells can be used to implement molecular codes, i.e. realises contingent mappings. For example, the code (Table D.1, no. 28) allows to map 2-Oxoglutarate (KEGG compound id: C00026) and L-Cysteine (C00097) to 2,4-Dihydroxyhept-2-enedionate (C06201) and N-Carbamyl-L-glutamate (C05829). The first molecular context contains pyrovate (C00022), L-glutamate (C00025) and water (C00001), and the second molecular context contains 4-aminobutanoate (C00334), succinate semialdehyd (C00232), NH4 (C00014) and also water (C00001). Since water is in both contexts it cannot be a determining factor of the mappings. This is especially true, because water, in principle, is present at every reaction in the cell. Figure 5.16 shows the approximate location of the participating species in the KEGG map of the metabolic network. If the cell could regulate the context, it could implements an encoded mapping between domain and codomain. Regulation could be for example on concentration level. Such codes can be characteristics of internal signalling, e.g. the implementation of molecular sensors (cp. [110]). Using enzymes (which were not part of the used model) enables the cell to regulate its reactions much better. For future studies enzymes should be included in the network to obtain a more detailed code analysis.

5.9.2 Cellular signal transduction

Cells maintain different systems for signal transmission and integration [111]. The transduction of molecular signal across the membrane can be understood as a molecular code. From a theoretical perspective the mapping from extracellular first messengers to internal second messengers is a molecular code mediated by the plasma membrane receptors. In general, signal transduction fulfils the properties of Barbieri's organic codes, since external signals like hormones in humans, or acyl-homoserine-lactones in gram-negative bacteria are from a different chemical world than the internal second messengers, like cyclic AMP, or other internal signal transmission systems, like phosphorylation cascades. The association between these two world is realised by receptor proteins located in the cell's membrane. The receptors perform two recognition steps: The first recognises the signal at the extracellular side, the second recognition process acts on the cytosolic side

[3]Kyoto Encyclopedia of Genes and Genomes, http://www.genome.jp/kegg/

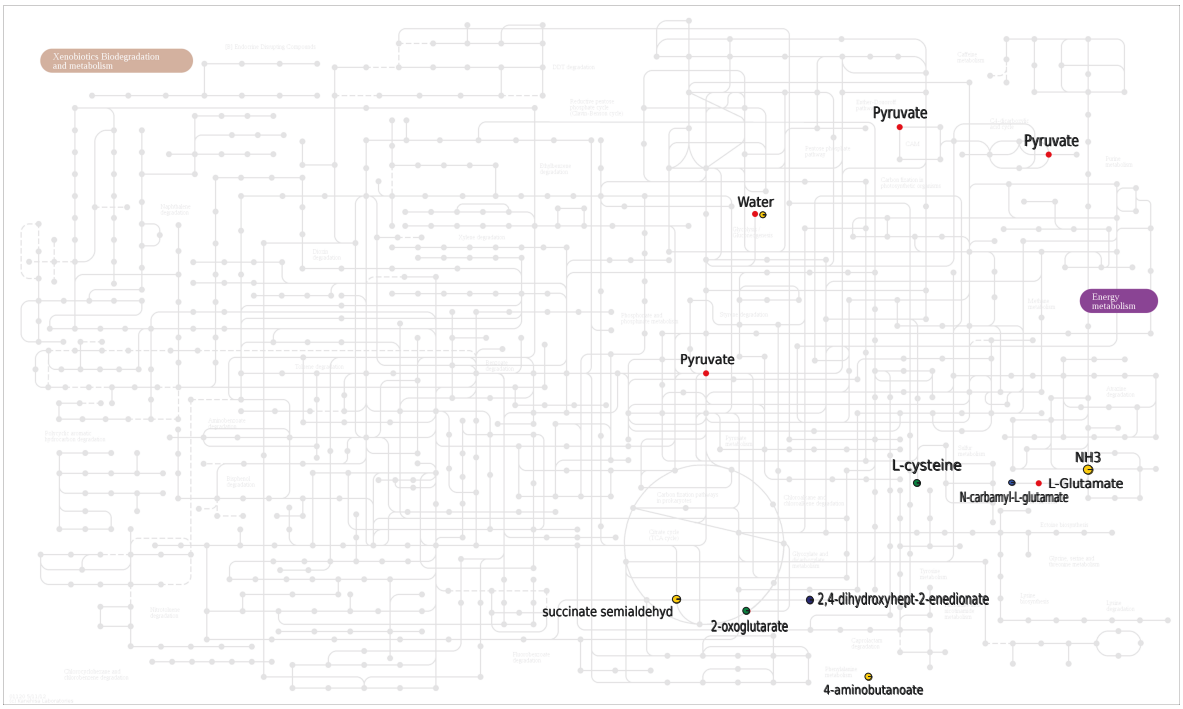


Figure 5.16 Metabolic map of the KEGG network. Map of the metabolic network obtained from the KEGG database showing the approximate positions of the components of code 28 (cf. Table D.1). Domain - green, codomain - blue, context 1 - red, context 2 - yellow.

and leads to the production of second messengers or the signal transmission by other processes, e.g. activation of proteins by phosphorylation. Due to the modular structure of many receptor protein complexes it can be assumed that the relation between a signal and the intracellular signalling is (to some extend) arbitrary and a code is instantiated. I will here analyse a network model of the known human signal transduction mechanisms. The model includes signalling by epidermal growth factor [112], fibroblast growth factor, insulin receptor [113], nerve growth factors [114], platlet-derived growth factor [115], vascular endothelial growth factors [116], stem cell factors [117], phospholipase C-γ mediated signalling [118], AKT signalling [119], the RAF/MAP kinase cascade signalling [120], Rho GTPases [121], bone morphogenetic protein pathway [122], TGF beta signalling [123], NOTCH signalling , the G protein coupled receptor receptors [124], Wnt signalling [125], the Hippo pathway [126], and the integrin cell surface interactions [127]. The complete network was obtained from the Reactome database (identifier: `REACT_111102.2`) [128]. All major signalling mechanisms known from human cells are included in the model making this reaction network a promising candidate to identify molecular codes using our algorithms. The network contains 1725 molecular species and 922 reaction rules (Figure 5.17). The network is structure in a large number of subnetworks, some only representing special ligand binding processes (lower part of Figure 5.17). A large module containing the integrin signalling (upper right corner) and a large module (center) that contains all other signalling processes. The very center contains ATP which is involved in a very large number of reactions. The dense structure of the network suggests also that crosstalk between different pathways is modelled. The molecular species in the network model represent single proteins, or other components,

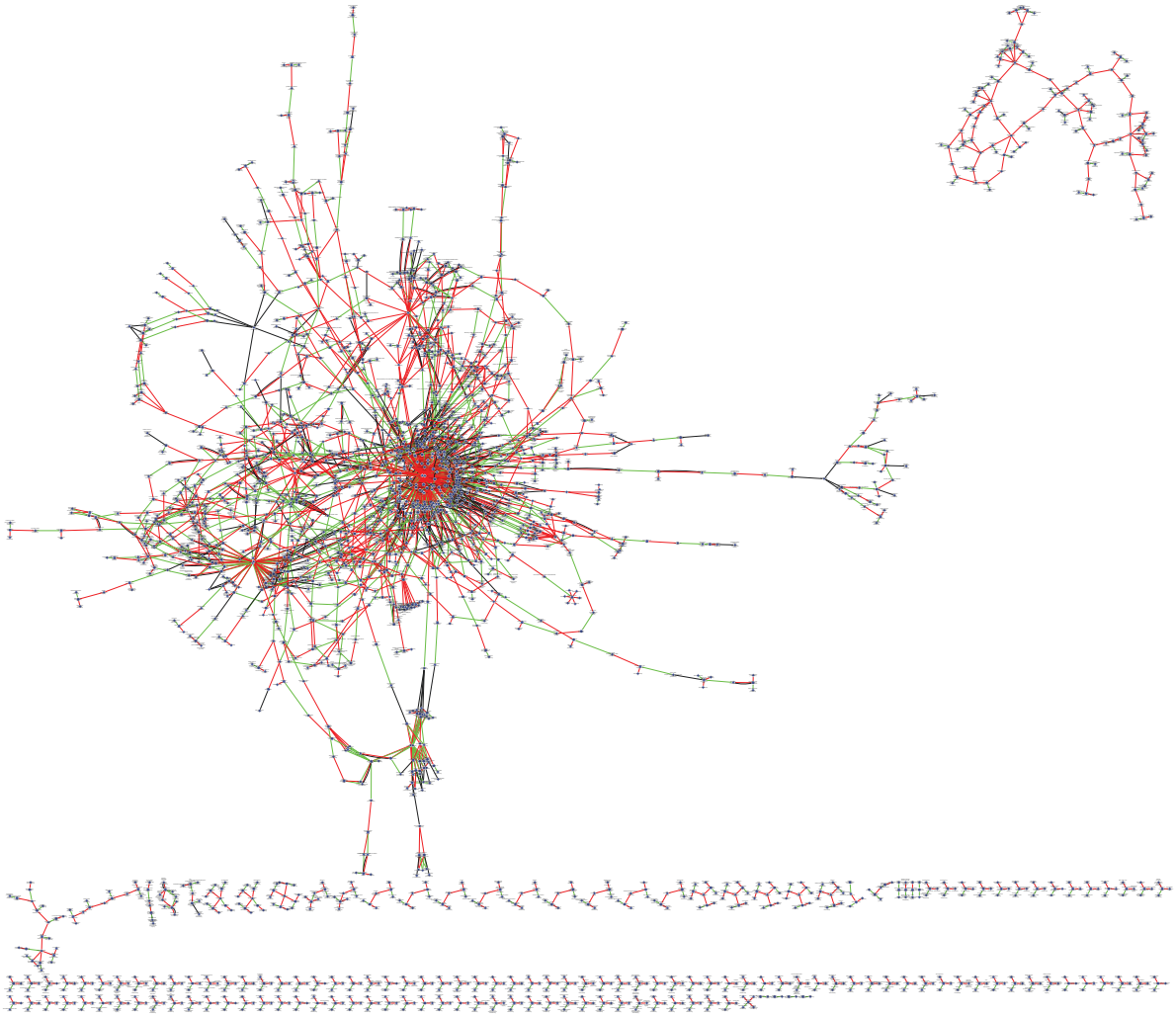


Figure 5.17 Reaction network of the human signal transduction (REACT_111102.2, www.reactome.org). The network shows all molecular species and reaction of the reactome model of signal transduction.

but also can stand for general families of molecular species, e.g. the species "GPCR that activates Gi[plasmamembrane]". Species' intracellular localisation is given by the tags "[plasmamembran]","[cytosol]", and "[extracellular]", or combinations thereof in the case of some complexes.
To analyse this network I use, due to the large size, the Monte-Carlo subnetwork sampling heuristic. For this network a reasonable subnetwork size (50), a small value of K (1) and a coverage rate of 100000 empirically proved suitable. The algorithm results in 558 binary molecular codes.
I defined seven biological roles to access the codes structures: cofactors (COF) for all proteins of other molecules necessary for the signalling, but which are not actively participating, effectors (EFF) like adenylate cyclase that produces second messengers, ligands (L), receptors (R), ligand receptor complexes (LR), activated receptors (AR), molecules and proteins that just transmit the signal (ST) and second messengers (SM). Table C.1 in Appendix C summarises which molecular species have been identified in which semiotics role (sign, meaning, context) among the codes and also gives information about the assigned role.

Analysis of the participating molecular species First I analysed if the identified molecular species occur either exclusively as signs (meanings) or if multiple roles can be taken by a species. Therefore, I determined the indicator variables $I_s(a), I_m(a) \in \{0, 1\}$, i.e. a molecular species a participated in at least one code either as sign or as meaning, respectively. Table 5.10 shows the contingency table of the two variables. A χ^2 test on the data show no significant dependency between the two groups "used as sign" and "used as meaning" ($\chi^2 = 2.12, p = 0.146$).

Table 5.10 Contingency table of biological roles of participating molecular species.

$I_s \backslash I_m$	0	1
0	146	42
1	31	15

For the analysis of the identified molecular species that could participate in a code I counted the number of codes for each species where it can act either as sign, meaning or context. Many of the identified molecular species (146 of 234) are neither used as sign or meaning, but only as context. One third of the species (73) is used exclusively either as sign or meaning and only 15 species are used as both. The molecular species that can function as sign and meaning in different codes are classified mainly as signal transducing species (10), but also as ligand receptor complexes (6) and cofactors and effectors (1 each). All molecular species are complexes involving GTP or GDP, and GDP itself, which can also be used as sign and meaning.
Table 5.11 shows the results of the analysis of the biological role versus the semiotic role. The analysis of the medians shows that over all biological roles many of the molecular species are never used as signs or meanings (medians are zero) but more often as context. This is due to the higher proportion of molecular species that can act only in contexts. The analysis of the means is qualitatively the same (context > sign, meaning), but differs in the actual values. A further statistical analysis, e.g. to identify differences between the biological roles, seems not very promising on this dataset, because t-tests on the means can not be applied due to non-normality of the (empirical) distributions

and also non-parametric rank-test (e.g. a U-Test) are not very powerful here, because the medians are very similar (many zeros).

Table 5.11 Number of codes per semiotic role for the biological roles.

		median number of codes			mean number of codes		
biological role	N	signs	meanings	context	signs	meanings	context
AR	20	0	0	2.5	7.8	2.15	5.65
LR	57	0	0	4	4.89	7.46	15.30
R	23	0	0	1	1.47	2.43	3.39
L	5	0	1	0	3.6	5	3.6
ST	96	0	1	4	7.5	7.76	40.56
SM	1	0	0	5	0	0	5
COF	20	0	0	7	12.5	2.15	29.25
EFF	12	0	0.5	8.5	4.25	1.91	33.67

Analysing the code structures Structurally, the identified codes are not as expected, mainly between external ligands and internal second messengers (classical code), but can be found in any combination of biological roles, also with receptors as meanings for example. The most abundant combination are codes where a ligand bound receptor and a signal transduction molecule can be mapped to two signal transduction molecules. The second most abundant combination is similar, but maps a signal transduction molecule and a ligand bound receptor to a signal transduction molecule and a ligand bound receptor. This is a combination where a receptor is a meaning. If these codes could be really used by cells, e.g. for any kind of internal controls can be only determined by a dynamic validation. Table 5.12 summarises the combinations of biological roles that have been found together in a code sorted by abundance.
A proper validation, e.g. by expert knowledge and dynamical arguments, is necessary to identify the feasible codes which might lead to a reduced set of molecular codes.
A different set of parameter values for the algorithmic code identification certainly would result in a larger number of BMCs.

Table 5.12 Combinations of biological roles occurring together in codes.

Signs		Meanings		#codes	Signs		Meanings		#codes
ST	LR	ST	ST	69	R	AR	ST	LR	3
ST	LR	ST	LR	60	R	AR	LR	EFF	3
LR	COF	ST	ST	38	COF	LR	AR	ST	3
ST	ST	ST	ST	28	COF	COF	LR	ST	3
COF	ST	ST	SR	24	ST	ST	COF	ST	2
ST	ST	LR	ST	22	ST	COF	ST	R	2
AR	ST	ST	ST	19	ST	COF	LR	R	2
COF	ST	LR	ST	19	ST	COF	COF	ST	2
COF	AR	AR	LR	19	ST	COF	COF	COF	2
ST	LR	R	ST	14	LR	ST	ST	COF	2
ST	LR	L	ST	14	COF	COF	ST	R	2
COF	AR	LR	LR	13	COF	COF	ST	LR	2
ST	AR	LR	LR	9	ST	ST	COF	LR	2
AR	ST	LR	ST	9	ST	COF	COF	LR	2
AR	ST	AR	LR	9	ST	AR	ST	AR	2
COF	AR	LR	R	9	COF	AR	ST	AR	2
ST	R	ST	ST	8	AR	COF	ST	R	2
ST	L	ST	ST	8	ST	R	COF	ST	1
ST	AR	ST	LR	8	ST	L	ST	LR	1
AR	ST	ST	LR	8	ST	LR	ST	COF	1
ST	LR	COF	ST	7	ST	L	COF	ST	1
R	AR	LR	LR	7	ST	AR	LR	R	1
COF	LR	LR	ST	7	ST	AR	LR	AR	1
COF	COF	ST	ST	7	ST	AR	A	COF	1
ST	R	LR	ST	7	R	ST	LR	COF	1
LR	COF	COF	ST	7	LR	COF	AR	ST	1
COF	AR	ST	LR	5	LR	AR	LR	ST	1
ST	L	LR	ST	4	L	COF	ST	ST	1
AR	ST	ST	R	4	COF	COF	LR	R	1
AR	ST	ST	L	4	COF	COF	COF	ST	1
AR	ST	ST	COF	4	COF	AR	ST	ST	1
ST	AR	ST	R	4	COF	AR	LR	COF	1
LR	COF	R	ST	4	COF	AR	AR	AR	1
LR	COF	L	ST	4	AR	ST	AR	AR	1
ST	ST	R	ST	3	AR	COF	LR	LR	1
ST	ST	L	ST	3	AR	COF	LR	AR	1
LR	ST	LR	COF	3	AR	COF	COF	ST	1
LR	COF	LR	ST	3				Sum	558

Abbrev.: AR - activated receptor, COF - cofactor, EFF - effector,
LR - ligand bound receptor, L - ligand, R - receptor, ST - signal transducer.

5.10 Summary

This chapter showed the results of the application of the code identifying algorithms on various systems.

From random reaction reactions to a statistical null model I studied random reaction networks to learn a null model for molecular codes. Therefore, I generated random networks of different sizes and densities and applied the code identifying algorithms on the networks. For a fixed network size the resulting mean semantic capacity can be modelled as random variable over the density. The unimodal behaviour of the data suggested a unimodal probability distribution as basis for the model. I tested a normal, a log-normal and a gamma distribution. The distribution's parameters have been estimated from the empirical determined mean and variance. For each fit I calculated the goodness of fit using the euclidean distance between the data and the model's prediction. The gamma distribution showed the best fit over all sampled network sizes. Nevertheless, a prediction out of the range of the data is not possible, because the distribution's shape changes rapidly and, thus, cannot be used as model for network sizes

larger than 25. For a prediction of the semantic capacity for network sizes in the range of the data the model is well suited. The very basic approach to generate random networks can be extended by, for example, generating random network maintaining some network properties, e.g., node degrees (in/out), or reconstitute the order distribution of the contained reactions. Also the identified codes have not been filtered, for example, for core codes, thus the exponential growth of the code pairs may be a result of such effects.
The general fact that codes can be found also in random networks can also be interpreted with respect to the evolution of codes. It shows, that by random variation of the reaction network potential codes can be introduced into a system. To really use a code the system need to be able to regulate the code's context, either dynamically or on an evolutionary time scale (cf. Chapter 6).

Combustion chemistries and biological networks The analysis of a set of combustion chemistries supported the hypothesis that the implementation of arbitrary mappings may be an exclusive feature of biological systems. None of the analysed combustion chemistries contained codes. This result is strengthened by the fact that these networks are considered to be complete in the sense that all reactions that could happen among the contained molecular species are contained in the network model.
The analysed biological networks all (beside the merge of the genetic codes) have been obtained by a knowledge based approach, i.e. the reactions have been modelled based on expert knowledge about the system. This approach has been chosen, because network models from database are not complete, firstly, because scientific progress in the respective field does not yet yield complete model, and secondly, because biological systems only realise one of the potential mappings. In the latter case also more effort in research might not result in the detection of the specific reactions necessary for the code based analysis. The analysis of the merge network of known genetic codes shows that merging networks may be a suitable approach to acquire suitable network models. Such merging needs to be done carefully. It may only make sense if network models from the same environmental context are merged, like the genetic codes.
The detailed analysis of the gene translation systems of cells showed that depending on the level of detail of the model the results of a code based analysis can be different. Here, the additional modelling of the amino acyl synthetases increased the semantic capacity of the system.
The coupling with the gene regulatory network, which is also a highly semantic system on its own, showed how a meaning-sign-linkage effects the semantic capacity.
I analysed simple protein assembly networks and showed that in general codes can be formed with such systems, but dissociation can destroy this property. A detailed analysis of an actual biological protein assembly network, as for example in kinetochore assembly may be a promising target for further research so see whether the influence of dissociation is also important in real systems.

Large scale biological systems The analysis of large scale biological systems showed that also in network models derived from experiments codes can be found. I demonstrated that without a subsequent validation of the codes no proper estimation of the semantic capacity can be given. The huge amount of potential codes, either due to a strong fan-in/fan-out (cf. code nesting, Section 3.4) or the large network sizes lead to

Table 5.13 Semantic roles in the analysed biological systems.

#	system	domain	codomain	context
1	gene translation (GC)	DNA triplets	amino acids	tRNAs
2	gene translation (incl. synth.)	DNA triplets + tRNA	amino acids	tRNA + synthetases
3	gene regulation (GRN)	TF	gene product	genes
4	GRN + GC	TF + transcript	proteins	genes + tRNAs
5	phosphorylation casc.	concentration of initial kinase	concentration of target species	casc. pathway
6	protein assembly	protein complexes	protein complexes	protein complexes
7	signal tranduction (Reactome)	various	various	various
8	metabolism (KEGG)	various	various	various

Annotation of the code based analysis of the biological systems. In different systems different molecular species can function in different roles. The same species (e.g., genes) can have different roles in different codes. Abrev.: GC - genetic code, GRN - gene regulatory network, DNA - desoxyribonucleic acid, TF - transcription factors.

codes that are difficult to interpret. The validation using for example dynamics could help to reduce the number of codes to the feasible codes.

Code linkages lead to systems of codes The concept of code linkage allows to make the notion of interdependent codes as presented in [32] more precise. The schema presented in ([32, Fig.7, p.922]) illustrates linked codes (cf. Figure 5.18).
Here, I discuss how external signals are mapped to internal signals via a signal transduction code, internal signals mapped to gene transcripts via a gene regulatory code, and gene transcripts mapped to proteins via the genetic code. In [32] we classified the codes as signalling, manufacturing and operating semiosis, respectively, following [129]. Using the notion of code linkage we can now see that all these linked codes are MSCL-type linkages. Manufacturing semiosis is given when a semiotics process (a code) produces something, e.g. meaningful molecular species[129]. Signalling semiosis, on the other hand, "[creates] specific signalling associations between pre-existing objects" and "[and does] [..] not bring these objects into existence." [129]. Operating semiosis is present if "[..] a code-based generation of signals control[s] the working of another code ."[32]. In principle all forms of semiosis can be linked by the proposed linkage types (cf. Section 3.4.3) Empirically, signalling and manufacturing semiosis seem to correspond with MSCL-type linkages, while operating semiosis probably more strongly corresponds with MCMC-type linkages, but MSCL is also possible for epigenetic codes.
In the case of signal transduction the second messenger triggers some response in the cell by subsequent biochemical reactions, e.g. phosphorlylations, i.e. there exists a path through the network leading to a sign of the linked code. This target molecular species very likely is a transcription factor and as such part of a subsequent (linked) code, here the gene regulatory code. The signalling semiosis of the signal transduction code thus is directly linked to a signalling semiosis of the gene regulatory code. The gene regulatory code, governs the mapping between transcription factors and gene products (e.g. mRNA), by this it can be also classified as signalling semiosis, since it copies the proper information of the DNA into mRNA. The mRNA is then translated by executing the genetic code and produces a protein. The genetic code can be classified as manufacturing semiosis. Between gene regulatory code and genetic code also a MSCL-linkage exists, since the mRNA contains the signs of the genetic code. The complete chain has the length 3. The epigenetic codes very likely have an effect on the execution of other codes, i.e. they are regulating these codes. This effect can be realised by the

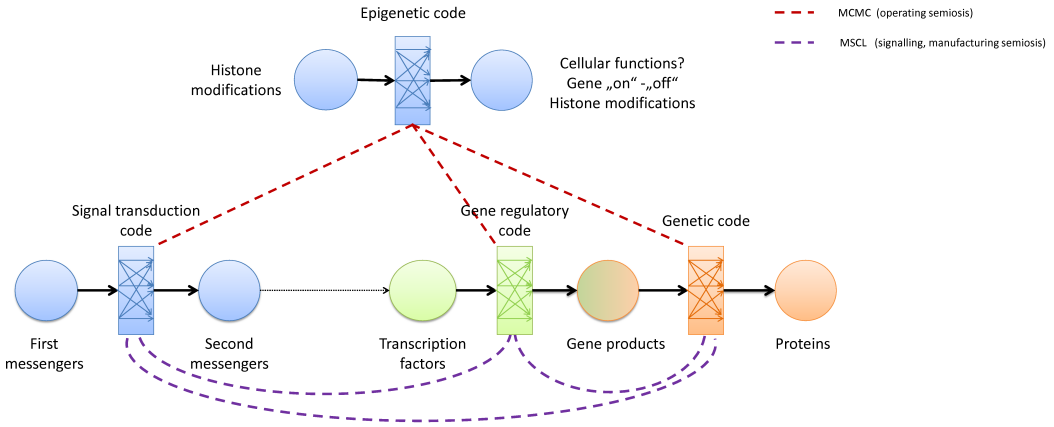


Figure 5.18 A system of codes emerging from code linkages. Circles denote set of molecular species. Boxes denote molecular codes. Solid arrows connect set of species and codes. The dotted arrow between second messengers and transcription factors stands for a variety of cellular processes involved in intracellular signal transmission. The dashed red and blueish lines indicate the two code linkages types MSCL and MCMC. The epigenetic code is hypothesised to control other codes by the MCMC-type linkage. The other codes are related by a MSCL-type linkage. The linkage types can be aligned with the notion of signalling, manufacturing and operating semiosis. MSCL-linkages from the meanings of the epigenetic code towards other codes have been omitted here. (Adapted from [32]).

linkage of an epigenetic meaning to the context of another code. I here hypothesise that such links can be found between the epigenetic codes and the other know molecular codes realised by cells. Novel discoveries in any of these cellular subsystems may alter and extend the picture I sketched here. Especially research in the histone code will give many new insights in the future.

Chapter 6

Towards pragmatics

In this chapter I want to present a number of ideas that lead from the pure semantic, structural level to the dynamic, pragmatic level of molecular codes.

6.1 Code validation

Dynamic code validation The analysis of photochemistries (Section 5.4), metabolism (Section 5.9.1) and signal transduction (Section 5.9.2) demonstrated that algorithmically identified molecular codes may need some kind of subsequent validation. This validation is necessary, because the definition, and thus the algorithms, neglects the dynamics of the system. In the case of the photochemistries light could have been used for an encoded mapping (structural information), but because this was in the night-side model no light should be present (dynamic information). For metabolism and signal transduction for many of the found codes dynamics may lead to non-injective mappings, i.e. when both contexts are realised simultaneously. Generalising this idea leads to the notion of *code validation*.
Given a molecular code f and all its alternative mappings g_i we say f is *valid* if at any time interval in the system's time course

1. all elements of one molecular context of f are present either simultaneously, or sequentially.

2. no two alternative contexts are present at the same time.

The definition basically requires that in the dynamic execution of the system the molecular context should be present in such a manner that the mapping can be executed and that non of the other mappings can be executed simultaneously, to obtain a unique mapping. Algorithmically, this can be checked by computer simulations. We can also use this information to adjust the semantic capacity to the number of valid codes.
The validation step could also be performed in wet-lab experiments. The definition of such experiments could orientate basically at the formulation of the code-identifying algorithms, i.e. mixing one sign and the molecular context should result in the presence of one respective meaning. The advantage of an experimental validation, especially *in-vivo*, is that the system is complete, and thus wrongly identified codes, e.g. because of incomplete network models, can be ruled out.

Code probability Beside the dynamic and experimental validation, more generally, we can try to calculate a code's probability.

Relevant questions in this regard are:

1. Given a molecular code which of the alternative mappings has the maximal probability. under realistic assumptions?

2. Given a set of molecular codes of a network, which code has the maximal probability, under realistic assumptions?

Both questions are similar, but tackle different aspects of a system's semantic capacity. Question (1) asks for the probability of one (unique) realisation of a code, i.e. which alternative context is chosen (cf. code determination in the next section)? Question (2) focusses on the overall semantic capacity.
Answering these questions only makes sense under realistic conditions. By this I mean that all relevant parameters, like kinetic rates, temperature, pH, concentrations, just to mention a few, have to be modelled in realistic ranges. In general, each mapping's $f : A \overset{C}{\mapsto} B$ probability $P(f)$ can be defined via the implied reactions $\rho \in R$ leading to

$$P(f) = \prod_{\rho} P(\rho).$$

Assuming a well-stirred reaction vessel the probability of reaction ρ to fire is given by

$$P(\rho) = P(\text{reactants collide}) \cdot P(\text{reactants react on collision}).$$

The probability for a collision is given by the reactants concentrations, while the probability that the reaction happens can be any constant or dependent on the actual reactants.
A s-t path's p_s^t probability is given by

$$P(p_s^t) = \prod_{\rho \in p_s^t} P(\rho).$$

For a dynamic framework time can also be included in the probabilities, i.e. reactants need not only to be in vicinity to each other, but also be present at the same time.
For a BMC between $\{A_1, A_2\}$ and $\{B_1, B_2\}$ there exists the four paths $p_{A_1}^{B_1}$, $p_{A_1}^{B_2}$, $p_{A_2}^{B_1}$, $p_{A_2}^{B_2}$. A unique mapping is given in the determining cases that the probabilities $P(p_{A_1}^{B_1}) + P(p_{A_2}^{B_2}) = 0$ (implying that $P(p_{A_1}^{B_2}) + P(p_{A_2}^{B_1}) \neq 0$) or $P(p_{A_2}^{B_1}) + P(p_{A_1}^{B_2}) = 0$. In all other configurations there exists a non-zero probability that the two alternative mappings are realised simultaneously. If the probabilities of the paths of the alternative mapping are very low they can be neglected. If not the mapping is no code, because the mapping is not unique. Because the probabilities can be estimated from the actual system a code can be validated to some extend. Suitable thresholds to decide whether a mapping can be used as code and which of the alternatives is chosen needs to be determined empirically, e.g. by taking into account reaction rates for actual chemical reactions.
Knowing the probabilities p_i of all molecular codes f_i allows to recalculate the network's semantic capacity by weighting each code with its probability. Assuming all molecular codes are independent of each other, the realisation of f_i does not change the probability to realise f_j, for $i \neq j$. Then Sc can be calculated by

$$Sc_p(N) = \prod p_i,$$

giving the probability that N realises all identified molecular codes. Imagine we identified ten molecular codes each with an equal probability of 0.5. Then the semantic capacity, the probability that all codes are realised, would be $0.5^{10} \approx 0.001$.
Actually, the current understanding of the evolution of biological systems and codes very strongly speaks against a complete independence of molecular codes. On the basic level codes always are grounded in the molecular species, that either needs to be produced or taken up by the system. So the execution of the code needs other processes that are regulated, probably by a different code. As soon as any kind of dependency exists between two codes, e.g. nested codes, and MSCL or MSMC linkages (see Section 3.4), the calculation gets more complicated and needs further research.

6.2 Code determination

The static description of potential codes does not guarantee that the cell can use this set-up for encoding information. Thus, cells need to guarantee that the alternative codes are not realised together, to unambiguously use the code for information transfer. So, on the pragmatic level cells have to "choose" which of the two mappings are preserved to guarantee that a distinction between the signs can be made. There does exist three pathways to guarantee the uniqueness of the mapping:

- *evolutionary choice* - denotes the process that one of the alternative codes is fixed in a evolutionary sense, i.e., the other codes are not maintained in the same system.
- *time separation* - denotes the effect that cells can switch between the alternative mappings by regulating the paths from signs to meanings on short to medium time scales (not evolutionary). By this cells are very flexible in their mappings, e.g. to react to changing environmental or internal states.
- *compartmentalisation* - allows for the simultaneous realisation of the codes. By separating the codes in different compartments the uniqueness of the mapping is maintained.

All three paths (Figure 6.1) can be observed in actual biological systems. Please note that these are not necessarily disjoint concepts. Compartmentalisation can happen in one cell where different mappings are realised in different compartments of the cell. But also the selection of codes in different species can be seen as compartmentalisation, where the different cells are the compartments. Then, if the other code cannot be realised by the other cell it is also an evolutionary choice. Both processes occur at least in the genetic code where different codes are implemented in different species [84]. Time separation can be understood as a regulated switch of mappings, e.g. in mitotic control where the presence of a protein called Cdc20 inhibits the Anaphase-Promoting Complex (APC) during the activated spindle assembly checkpoint (SAC), while in the context of the inactivated checkpoint, Cdc20 activates APC [130, 131].
The (evolutionary) choice between the alternative mappings depends on various factors, e.g. the chemical properties of the system, or the coevolution history of the chemical system. Other factors could be the (metabolic) cost for maintaining certain pathways.
Suitable models need to be developed to analyse the evolution of molecular codes on the network level properly. The simulation of the evolution of networks, or analyses using evolutionary game theory might give more insights into this topic.

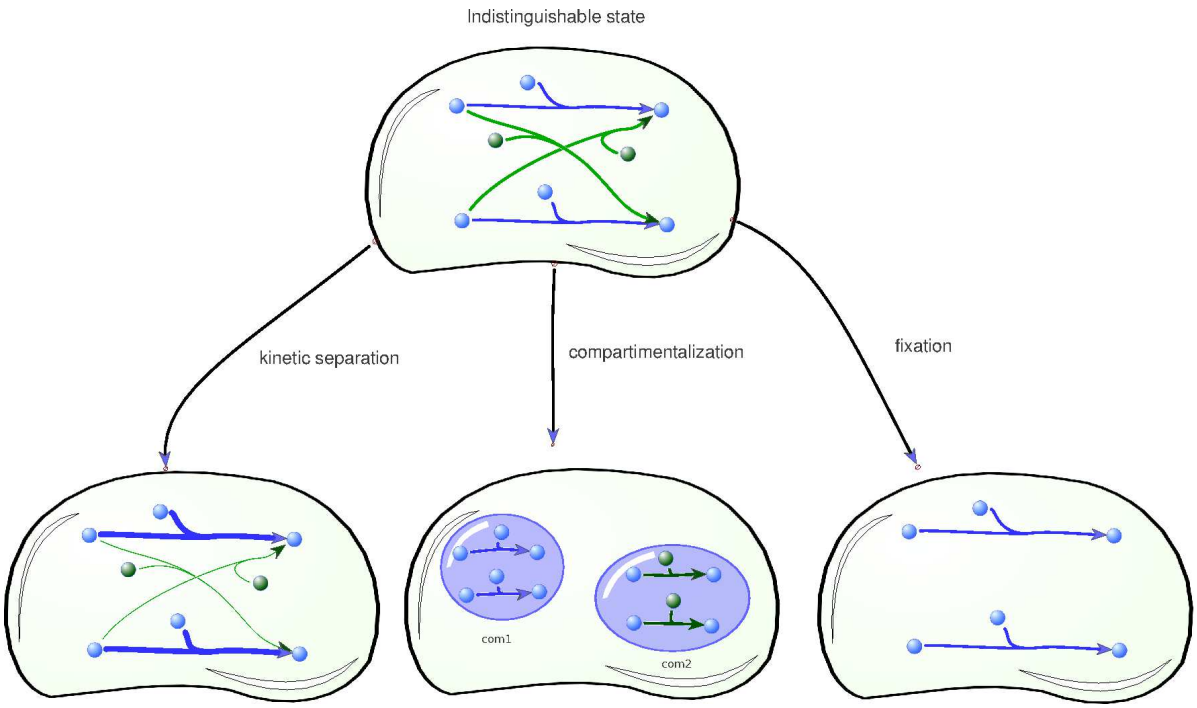


Figure 6.1 Illustration of the three pathways of code determination. Kinetic separation (left) leads to one of the mappings by increasing the rates of the reactions realising this mapping. Compartmentalisation (middle) separates the two mappings either in compartments, or different species. Fixation (right) deletes the alternative mapping completely.

6.3 Codes between system states

In the static framework only mappings between molecular species could be detected via the reactions of the system. Some codes can only be identified in a dynamic framework, as could be seen at the phosphorylation cascade example. Dynamics can be (re-)introduced to the network model via the kinetic laws of the reactions. The system's dynamic behaviour can be concisely modelled as the solution of a system of ordinary differential equations. Let $x_t = (x_{1,t}, x_{2,t}, \ldots, x_{|\mathcal{M}|,t})^T$ be a vector containing the concentrations of the system components at time point t. The systems behaviour is determined by $\frac{dx_t}{dt} = f(x_t)$. Using time, the network structure, and the kinetics as causal relationship between two system states is will be possible to define a (dynamic) molecular code that maps from a state $x^{(1)}$ to a state $x^{(2)}$. A code, in analogy to the static code definition, is present, if under changing contexts the mapping changes. A context, here, could be for example the initial concentration, or the concentration level of a selected subset of molecular species $C = x^c$ the alternative context C' could now be a different set of species $x^{c'}$, or the same set with a different concentration vector x'^c. Also dynamic switching between mappings can be easily implemented, because the context can be part of the system, and thus, its concentration vector can easily be influenced by the general system's behaviour (unless its an uncorrelated, separated subsystem).

The introduction of dynamics opens a huge new chapter in code biology. When system states can be used a signs and meanings (if the cell can read the state somehow) then the model could also describe information transfer by dynamic behaviour, e.g. calcium oscillations. Then mapping can also be realised between fixed points of the system, or any

Table 6.1 Comparison static vs. dynamic code concept.

property	static framework	dynamic framework
entities	molecular species (present/absent)	molecular species (concentration levels), system states (fixed points, attractors)
mapping realised by	reactions, paths	reactions, paths, kinetics, time
code identification	network pattern	behaviour in state space
analysis	number of codes, code relations	number of codes, code relations, information theory, dynamical systems theory (stability, etc.)

kind of complex attractors. Additionally, the whole toolbox of dynamic system analysis gets available for a code based analysis of systems. Table 6.1 gives a general overview of the conceptual differences between the static and the dynamic approach. Basically, the static approach is a special case of the dynamic framework with a threshold operation on the concentrations. The dynamic framework will be much harder to analyse, but it probably can explain more phenomena, e.g. calcium waves, and is accessible to the toolbox of dynamical systems theory.

Chapter 7

Discussion and Outlook

I developed a formalisation of molecular codes in the context of reaction network models. This thesis covered the conceptual introduction to codes and discussed the usage of the term code for different biological systems. I also developed different algorithms for code identification and presented the results of the algorithms' application to various systems (discussed at the end of Chapter 5).
Many open questions and ways to continue research in this field remain.

Improvement of algorithms The presented algorithms follow brute-force strategies. For the pathway-based algorithm I suggested two improvements, first, a parametrisation on the K-shortest paths and, second, a Monte-Carlo type sampling algorithm. Both allow for the analysis of larger networks but in practical situations do not find all codes. Additionally, a computational challenge remain, because the runtime complexity (number of paths, number of closed sets) leaves the feasible problem sizes quite fast. Thus a need for improved methods is still given.

Choice of network models The code based analysis of systems needs complete network models, i.e. the network is required to represent all possible reactions that can happen among the molecular species and thus is a complete model of the world. In this thesis I showed that such networks can partly be reconstructed by expert knowledge or merge approaches. The knowledge based approach is especially necessary if the hypothesis that cells maintain only one of the potential mappings is true. Then, networks derived from experiments cannot contain the alternative mappings, because they are invisible to experimental techniques. If different mappings are implemented in different compartments a merge on the reaction networks can help to bring both realisations together in one network. This has been demonstrated in Section 5.5 for the genetic code. Data sources like the Biomodels database (http://www.ebi.ac.uk/biomodels-main/) usually contain only subnetworks that does not reflect the complete system, but only explain certain selected subsystems. Large scale network models like KEGG or BioCyc converge towards complete models, but may contain faulty data, even with constant curation. Additionally, a computational challenge remains, because the current algorithms can not, or only hardly handle such large networks. The proposed heuristics (K-shortest paths, Monte-Carlo sampling) does not guarantee to identify all codes and thus other alternative approaches needs to be identified.
Throughout the thesis it showed that slightly different models of the same chemistry can have an effect on the results of the code based analysis. For example, the night side model of Mars had codes when only taking the inflow reaction away, but no codes if

all reactions using light were deleted from the model. Increased detail in the network models can also lead to increased semantic capacity, as for example in the network model of the coupled GC-GRN network. The most detailed model might be best suited for a code based analysis, but will be hard to analyse. Thus, for practical applications a trade-off between level of detail and computational feasibility has to be found.

Evolution of Molecular Codes Many hypothesis have been made how the genetic code has evolved [132, 133, 17]. Koonin [17] stated that to understand the evolution of the genetic code we have to understand the evolution of codes in general. The codes defined in this paper may be suitable to understand how codes in general evolve. A working hypothesis emerging from the results presented in this thesis is that during the origin of life (chemical evolution) and the evolution of life the semantic capacity in the reaction systems discovered and incorporated by living systems increased. A basic, but not necessarily the best, measure for semantic capacity may be the number of BMCs as presented in this thesis. Possible other measures of semantic capacity (core codes, probabilities) have been discussed in this thesis. The hypothesis is supported by intrinsic differences in the subsystems used by cells. For example, the metabolic system is much more governed by the physical and chemical rules applied to the reactions (e.g. mass conservation) than the gene regulatory system whose semantic capacity is based in the contingent combination of promoters and protein encoding DNA. Nevertheless the metabolism could be used for encoding information if cells can regulate their metabolic pathways appropriately (cp. results presented in Section 5.9.1). The validation of the hypothesis needs careful integration of the data and further development of the algorithms.
Also in the context of evolution of codes it can be hypothesised that cost efficient codes are preferred over more costly codes. Costs, here, can be for example measured by metabolic costs of the paths realising a code. Tlusty [23] uses a different notion of costs based on the number of bits necessary to encode the transmitted information, assuming that more complex, and thus more expensive, signs are necessary for a larger information content. As have been shown by Tlusty a code itself has a fitness that is determined by its encoding properties [23]. Both notions of costs cover different aspects of a code. While the first notion is more directly linked to the energy the cell has to spend to maintain the mapping Tlusty's notion is more abstract on the properties of the signs (and meanings). Applying a fitness measure to a code, it can be understood to be relevant also to biological fitness. Now it can be hypothesised that a biological species' fitness depends on its capability to encode information.
If codes are beneficial for a species' fitness it can be also hypothesised that cells, in the course of evolution, increased the number of codes. Cells may have increased their semantic capacity by acquiring new biochemical subsystem that allowed for encoding information. Proving this hypothesis needs though even more research efforts, e.g. in establishing evolutionary game theoretical models.

Towards dynamics This work provided a first step into a deeper understanding of certain properties of molecular codes. The molecular code framework is well suited to describe the mechanistic properties of molecular codes, but lacks for example the dynamic level. The analysis of phosphorylation cascades demonstrated that codes that are based on concentration levels are not covered by the framework in the actual state. The extension to a dynamic formulation thus is one of the major research themes in this

field. First steps have been made, though (cp. [80]).
The extension to a dynamic framework of molecular codes integrates into already established analysis techniques and can be coupled with steady state analyses where fixed points or attractors are analysed. It may also prove beneficial to couple code based network analysis to a Petri net formulation. Petri nets have been successfully applied in modelling and analysis of biological networks [134, 135] and come with a well defined set of concepts for the structural and dynamical analysis that also can be linked to the notion of molecular codes.
It also needs to be checked how the code concept is related to the notion of chemical organisations [61]. Both concepts are related through the notion of closed sets and potentially there exist codes between organisation. If so, then a (bio-)chemical system could move between its chemical organisations in an arbitrary way defined by a molecular context.

Relation to information theory The definition of BMCs captures some semantical aspect of biological information. A common approach to information in biological systems is to equate information with correlation or mutual information between two random sources, e.g. the message and its environment [1]. High mutual information would also be necessary for BMCs, but is not sufficient. In other words, measuring a correlation or mutual information between two worlds does not necessarily imply that there is a code or a semiotic structure. In addition "arbitrariness" is needed, represented formally by the alternative context C'. Otherwise the mapping is based on direct physical causal relationship or a natural sign (cf.[16]).
If we already know that a molecular codes exist, e.g. identified by the presented algorithms, the information theoretic analysis between signs and meanings can be very informative about the nature of the code, and perhaps also helps in validating codes. To model molecular codes in information theoretic terms signs and meanings have to be understood as random variables, either discrete (on/off) or continuous. Then, also certain assumptions about the used distributions have to be made, or empirically determined, if possible. For entropy measures the empirical determination might be feasible, but for mutual information, which needs the joint entropy, the non-realised associations might never be measurable. Here only reasonable estimates can help.

Simulation environment The analysis of the pragmatic level of molecular codes can be implemented in the simulation framework ArtBact developed by Erbach [136] and Weisensee [137]. ArtBact allows for the evolution of cellular networks. Thus, it is well suited to tackle questions related to the structural evolution of molecular codes under defined environmental conditions.
More concretely, I suggest to perform an evolution experiment with two external chemoattractants, or other kind of signals. The bacterium contains two kinds of effector and should learn to transduce information about the external signal concentration via its regulatory networks to the effectors. The fitness in such an experiment can be a combination of biomass, i.e. the bacteria learn to survive, and, in a first step, the exclusive usage of one of the effectors. By this strong constraint we might be able to learn what kind of networks evolve to reach optimal fitness values. In particular, it might be interesting to see whether network structures similar to the formalisation of codes evolves, or if different approaches, e.g. by dynamic behaviour, get visible to get a higher fitness. The ArtBact framework allows to apply information theoretic measures like mutual

information to the generated time series data. This links the structural definition of molecular codes to dynamics and thus enter the pragmatic level.

Experimental validation Finally, the notion of codes directly generates input for potential wet-lab experiments. The codes identified in network models of a certain system can be checked by experiments that follow the closure algorithm. For a proposed molecular code the experiment needs to check whether for the two signs combined with the two contexts, independently, the two meanings are produced. The experimental validation of molecular codes is the best possible type of validation, because *in-vivo* the pragmatic dynamic level is always present and thus non-feasible codes can be identified exactly.
Overall, I presented a theoretical framework and demonstrated applications to various network models. As outlined in this chapter, the definitions with respect to chemical reaction networks opened the door to many new research questions that needs to be answered in future studies.

References

[1] C. Waltermann, E. Klipp (**2011**) Information theory based approaches to cellular signaling. *Biochim Biophys Acta*, 1810(10):924–932.

[2] T. Köhler, G. G. Perron, A. Buckling, C. van Delden (**2010**) Quorum sensing inhibition selects for virulence and cooperation in pseudomonas aeruginosa. *PLoS Pathog*, 6(5):e1000883.

[3] C. E. Shannon (**1948**) A mathematical theory of communication. *The Bell Systems Technical Journal*, 27:379–423, 623–656.

[4] G. Tkačik, A. M. Walczak (**2011**) Information transmission in genetic regulatory networks: a review. *J Phys Condens Matter*, 23(15):153102.

[5] P. Mehta, S. Goyal, T. Long, B. L. Bassler, N. S. Wingreen (**2009**) Information processing and signal integration in bacterial quorum sensing. *Mol Syst Biol*, 5:325. (doi:10.1038/msb.2009.79).

[6] T. Lenaerts, J. Ferkinghoff-Borg, F. Stricher, L. Serrano, J. W. H. Schymkowitz, F. Rousseau (**2008**) Quantifying information transfer by protein domains: analysis of the Fyn SH2 domain structure. *BMC Struct Biol*, 8:43.

[7] J. Monod (**1971**) *Chance and necessity*. Alfred Knopf, New York/NY. (Originally published 1970).

[8] B.-O. Küppers (**1990**) *Information and the origin of life*. MIT Press, Cambridge/MA. (Originally published 1986).

[9] C. Morris (**1971**) *Writing on the general theory of signs*. Mouton, Den Haag.

[10] P. Bralley (**1996**) An introduction to molecular linguistics. *BioScience*, 46(2):146–153.

[11] M. Gimona (**2006**) Protein linguistics - a grammar for modular protein assembly? *Nat Rev Mol Cell Biol*, 7(1):68–73.

[12] S. Artmann (**2008**) Biological information. In S. Sarkar, A. Plutynski (eds.), *A companion to the philosophy of biology*, no. 39 in Blackwell companions to philosophy, chap. 2, 22–39. Blackwell Publishing.

[13] A. Jayaraman, T. K. Wood (**2008**) Bacterial quorum sensing: Signals, circuits, and implications for biofilms and disease. *Annu Rev Biomed Eng*, 10:145–167.

[14] S. Artmann (**2007**) Computing codes versus interpreting life: Two alternative ways of synthesizing biological knowledge through semantics. In M. Barbieri (ed.), *Introduction to biosemiotics: The new biological synthesis*, 209–233. Dodrecht: Springer.

[15] D. Chandler (**2007**) *Semiotics: the basics*. Routledge, Abingdon, UK, 2nd edn.

[16] M. Barbieri (**2008**) Biosemiotics: a new understanding of life. *Naturwissenschaften*, 95(7):577–599.

[17] E. V. Koonin, A. S. Novozhilov (**2009**) Origin and evolution of the genetic code: the universal enigma. *IUBMB Life*, 61(2):99–111.

[18] S. Artmann (**2002**) Three types of semiotic indeterminacy in monod's philosophy of modern biology. *Sign System Studies*, 30(1):149–169.

[19] H. H. Pattee (**2008**) Physical and functional conditions for symbols, codes, and languages. *Biosemiotics*, 1(2):147–168.

[20] J. Maynard Smith (**2000**) The concept of information in biology. *Philosophy of science*, 67:177–194.

[21] S. Sarkar (**2000**) Information in Genetics and Developmental Biology: Comments on Maynard Smith. *Philosophy of Science*, 67:208–213.

[22] U. Stegmann (**2004**) The arbitrariness of the genetic code. *Biology & Philosophy*, 19(2):205–222.

[23] T. Tlusty (**2008**) Casting polymer nets to optimize noisy molecular codes. *Proc Natl Acad Sci U S A*, 105(24):8238–8243.

[24] J. Bierbrauer (**2005**) *Introduction to coding theory*. Discrete Mathematics and its applications. Chapman & Hall/CRC, Boca Raton, Fl.

[25] T. M. Cover, J. A. Thomas (**1991**) *Elements of Information Theory*. Wiley series in telecommunications. Wiley, New York, NY.

[26] S. Verdu (**1998**) Fifty years of Shannon theory. *IEEETransactions on Information Theory*, 44(6):2057–2078.

[27] D. J. MacKay (**2003**) *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

[28] T. Tlusty (**2008**) A simple model for the evolution of molecular codes driven by the interplay of accuracy, diversity and cost. *Phys Biol*, 5(1):16001.

[29] T. Tlusty (**2008**) Rate-distortion scenario for the emergence and evolution of noisy molecular codes. *Phys Rev Lett*, 100(4):048101.

[30] K. Vetsigian, C. Woese, N. Goldenfeld (**2006**) Collective evolution and the genetic code. *Proc Natl Acad Sci U S A*, 103(28):10696–10701.

[31] M. Barbieri (**2003**) *The organic codes: An introduction to semantic biology*. Cambridge University Press, Cambridge.

[32] D. Görlich, S. Artmann, P. Dittrich (**2011**) Cells as semantic systems. *Biochim Biophys Acta*, 1810(10):914–923.

[33] M. G. Safro, N. A. Moor (**2009**) Codases: fifty years after. *Mol Biol (Mosk)*, 43(2):230–242.

[34] L. L. Kiselev (**1990**) Aminoacyl-tRNA synthetases (codases) and their noncanonical functions. *Mol Biol (Mosk)*, 24(6):1445–1473.

[35] D. Schwarzer (**2010**) Chemical tools in chromatin research. *J Pept Sci*, 16(10):530–537.

[36] J.-S. Lee, E. Smith, A. Shilatifard (**2010**) The language of histone crosstalk. *Cell*, 142(5):682–685.

[37] S. J. Prohaska, P. F. Stadler, D. C. Krakauer (**2010**) Innovation in gene regulation: the case of chromatin computation. *J Theor Biol*, 265(1):27–44.

[38] A. Csordas (**1990**) On the biological role of histone acetylation. *Biochem J*, 265(1):23–38.

[39] B. M. Turner (**1993**) Decoding the nucleosome. *Cell*, 75(1):5–8.

[40] B. M. Turner (**2000**) Histone acetylation and an epigenetic code. *Bioessays*, 22(9):836–845.

[41] B. D. Strahl, C. D. Allis (**2000**) The language of covalent histone modifications. *Nature*, 403(6765):41–45.

[42] T. Jenuwein, C. D. Allis (**2001**) Translating the histone code. *Science*, 293(5532):1074–1080.

[43] K. A. Gelato, W. Fischle (**2008**) Role of histone modifications in defining chromatin structure and function. *Biol Chem*, 389(4):353–363.

[44] A. Lennartsson, K. Ekwall (**2009**) Histone modification patterns and epigenetic codes. *Biochim Biophys Acta*, 1790(9):863–868.

[45] B. M. Turner (**2002**) Cellular memory and the histone code. *Cell*, 111(3):285–291.

[46] S. Henikoff (**2005**) Histone modifications: combinatorial complexity or cumulative simplicity? *Proc Natl Acad Sci U S A*, 102(15):5308–5309.

[47] M. F. Dion, S. J. Altschuler, L. F. Wu, O. J. Rando (**2005**) Genomic characterization reveals a simple histone H4 acetylation code. *Proc Natl Acad Sci U S A*, 102(15):5501–5506.

[48] R. Margueron, P. Trojer, D. Reinberg (**2005**) The key to development: interpreting the histone code? *Curr Opin Genet Dev*, 15(2):163–176.

[49] J. Morinière, S. Rousseaux, U. Steuerwald, M. Soler-López, S. Curtet, A.-L. Vitte, J. Govin, J. Gaucher, K. Sadoul, D. J. Hart, J. Krijgsveld, S. Khochbin, C. W. Müller, C. Petosa (**2009**) Cooperative binding of two acetylation marks on a histone tail by a single bromodomain. *Nature*, 461(7264):664–668.

[50] J. I. Wu, J. Lessard, G. R. Crabtree (**2009**) Understanding the words of chromatin regulation. *Cell*, 136(2):200–206.

[51] H.-J. Gabius (**2000**) Biological information transfer beyond the genetic code: the sugar code. *Naturwissenschaften*, 87(3):108–121.

[52] H.-J. Gabius, S. André, H. Kaltner, H.-C. Siebert (**2002**) The sugar code: functional lectinomics. *Biochim Biophys Acta*, 1572(2-3):165–177.

[53] H.-J. Gabius (ed.) (**2009**) *The sugar code: Fundamentals of glycosciences*. Wiley-VCH, Weinheim.

[54] H. Rüdiger, H.-J. Gabius (**2009**) *The sugar code: Fundamentals of glycosciences*, chap. The biochemical basis and coding capacity of the sugar code, 3–13. Wiley-VCH.

[55] R. Laine (**1997**) The information-storing potential of the sugar code. In H.-J. Gabius (ed.), *Glycosciences: Status and Perspectives*. Chapman & Hall, London.

[56] J. Holgersson, A. Gustafsson, S. Gaunitz (**2009**) *The sugar code: Fundamentals of glycosciences*, chap. Bacterial and viral lectins, 279–300. Wiley-VCH, Weinheim.

[57] H. Rüdiger, H.-J. Gabius (**2009**) *The sugar code: Fundamentals of glycosciences*, chap. Plant lectins, 301–315. Wiley-VCH, Weinheim.

[58] H.-J. Gabius (**2009**) *The sugar code: Fundamentals of glycosciences*, chap. Animal and human lectines, 317–328. Wiley-VCH.

[59] N. Sharon, H. Lis (**1989**) Lectins as cell recognition molecules. *Science*, 246(4927):227–234.

[60] D. Görlich, P. Dittrich (**2011**) Identifying molecular organic codes in reaction networks. In G. Kampis, I. Karsai, E. Szathmáry (eds.), *Advances in Artificial Life. Darwin Meets von Neumann*, vol. 5777 of *Lecture Notes in Computer Science*, 305–312. Springer Berlin / Heidelberg.

[61] P. Dittrich, P. S. D. Fenizio (**2007**) Chemical organization theory. *Bull Math Bio*, 69(3):1199–1231.

[62] W. Fontana, L. W. Buss (**1994**) The arrival of the fittest: Toward a theory of biological organization. *Bull Math Bio*, 56:1–64.

[63] C. Meinel, M. Mundhenk (**2002**) *Mathematische Grundlagen der Informatik*. Teubner B.G. GmbH.

[64] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. L. Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita,

J. Wagner, J. Wang, S. B. M. L. Forum (**2003**) The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531.

[65] A. Finney, M. Hucka (**2003**) Systems biology markup language: Level 2 and beyond. *Biochem Soc Trans*, 31(Pt 6):1472–1473.

[66] B. Roberts, D. P. Kroese (**2007**) Estimating the number of s-t paths in a graph. *Journal of Graph Algorithms and Applications*, 11(1):195–214.

[67] J. Y. Yen (**1971**) Finding the K shortest loopless paths in a network. *Management science*, 17:712–716.

[68] D. Eppstein (**1998**) Finding the k shortest paths. *SIAM J on Computing*, 28(2):652–673.

[69] F. J. Planes, J. E. Beasley (**2008**) A critical examination of stoichiometric and path-finding approaches to metabolic pathways. *Brief Bioinform*, 9(5):422–436.

[70] E. Q. V. Martins, M. M. B. Pascoal (**2003**) A new implementation of yen's ranking loopless paths algorithm. *4OR: A Quarterly Journal of Operations Research*, 1:121–133. 10.1007/s10288-002-0010-2.

[71] E. Kaiser, T. Wallington, M. D. Hurley, J. Platz, H. J. Curran, W. J. Pitz, C. K. Westbrook (**2000**) Experimental and modeling study of premixed atmospheric-pressure dimethyl ether-air flames. *J Phys Chem A*, 104(35):8194–8206.

[72] N. M. Marinov (**1999**) A detailed chemical kinetic model for high temperature ethanol oxidation. *Int J Chem Kinet*, 31:183–220.

[73] M. O. Conaire, H. J. Curran, J. M. Simmie, W. J. Pitz, C. Westbrook (**2004**) A comprehensive modeling study of hydrogen oxidation. *Int J Chem Kinet*, 36(11):603–622.

[74] T. Turnyi, K. Hughes, M. Pilling, A. Tomlin (**2001**). The Leeds methane oxidation mechanism. online. Version 1.5, available at http://www.chem.leeds.ac.uk/Combustion/methane.htm.

[75] W. Banzhaf (**1993**) Self-replicating sequences of binary numbers. *Comput Math Appl*, 26:1–8.

[76] H. Nair, M. Allen, A. D. Anbar, Y. L. Yung (**1994**) A photochemical model of the martian athmosphere. *Icarus*, 111:124–150.

[77] F. Centler, P. Dittrich (**2007**) Chemical organizations in atmospheric photochemistries: a new method to analyze chemical reaction networks. *Planet Space Sci*, 55(4):413–428.

[78] F. Centler (**2008**) *Chemical organizations in natural reaction networks*. Ph.D. thesis, Friedrich-Schiller-Universität Jena.

[79] F. H. Crick, L. Barnett, S. Brenner, R. J. Watts-Tobin (**1961**) General nature of the genetic code for proteins. *Nature*, 192:1227–1232.

[80] J. DeBeule, E. Hovig, M. Benson (**2010**) Introducing dynamics into the field of biosemiotics. *Biosemiotics*, 4:5–24.

[81] R. Knippers (**2006**) *Molekulare Genetik*. Georg Thieme Verlag, Stuttgart, 9 edn. In German.

[82] S. Osawa, T. H. Jukes, K. Watanabe, A. Muto (**1992**) Recent evidence for evolution of the genetic code. *Microbiol Rev*, 56(1):229–264.

[83] T. H. Jukes, S. Osawa (**1993**) Evolutionary changes in the genetic code. *Comp Biochem Physiol B*, 106(3):489–494.

[84] A. Elzanowski, J. Ostell (**2010**). The genetic code. Last update: July 7, 2010. Retrieved: March 1, 2011.
URL `http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi`

[85] M. Di Giulio (**2008**) An extension of the coevolution theory of the origin of the genetic code. *Biol Direct*, 3:37.

[86] G. D. Clark-Walker, G. F. Weiller (**1994**) The structure of the small mitochondrial DNA of Kluyveromyces thermotolerans is likely to reflect the ancestral gene order in fungi. *J Mol Evol*, 38(6):593–601.

[87] H. Himeno, H. Masaki, T. Kawai, T. Ohta, I. Kumagai, K. Miura, K. Watanabe (**1987**) Unusual genetic codes and a novel gene structure for tRNA(AGYSer) in starfish mitochondrial DNA. *Gene*, 56(2-3):219–230.

[88] H. T. Jacobs, D. J. Elliott, V. B. Math, A. Farquharson (**1988**) Nucleotide sequence and gene organization of sea urchin mitochondrial DNA. *J Mol Biol*, 202(2):185–217.

[89] B. Batuecas, R. Garesse, M. Calleja, J. R. Valverde, R. Marco (**1988**) Genome organization of Artemia mitochondrial DNA. *Nucleic Acids Res*, 16(14A):6515–6529.

[90] S. Osawa, T. Ohama, T. H. Jukes, K. Watanabe (**1989**) Evolution of the mitochondrial genetic code. I. origin of AGR serine and stop codons in metazoan mitochondria. *J Mol Evol*, 29(3):202–207.

[91] J. R. Garey, D. R. Wolstenholme (**1989**) Platyhelminth mitochondrial DNA: evidence for early evolutionary origin of a tRNA(serAGN) that contains a dihydrouridine arm replacement loop, and of serine-specifying AGA and AGG codons. *J Mol Evol*, 28(5):374–387.

[92] T. Ohama, S. Osawa, K. Watanabe, T. H. Jukes (**1990**) Evolution of the mitochondrial genetic code. IV. AAA as an asparagine codon in some animal mitochondria. *J Mol Evol*, 30(4):329–332.

[93] R. J. Hoffmann, J. L. Boore, W. M. Brown (**1992**) A novel mitochondrial genome organization for the blue mussel, Mytilus edulis. *Genetics*, 131(2):397–412.

[94] G. A. Durrheim, V. A. Corfield, E. H. Harley, M. H. Ricketts (**1993**) Nucleotide sequence of cytochrome oxidase (subunit III) from the mitochondrion of the tunicate Pyura stolonifera: evidence that AGR encodes glycine. *Nucleic Acids Res*, 21(15):3587–3588.

[95] J. L. Boore, W. M. Brown (**1994**) Complete DNA sequence of the mitochondrial genome of the black chiton, Katharina tunicata. *Genetics*, 138(2):423–443.

[96] A. Kondow, T. Suzuki, S. Yokobori, T. Ueda, K. Watanabe (**1999**) An extra tRNAGly(U*CU) found in ascidian mitochondria responsible for decoding non-universal codons AGA/AGG as glycine. *Nucleic Acids Res*, 27(12):2554–9.

[97] M. J. Telford, E. A. Herniou, R. B. Russell, D. T. Littlewood (**2000**) Changes in mitochondrial genetic codes as phylogenetic characters: two examples from the flatworms. *Proc Natl Acad Sci U S A*, 97(21):11359–11364.

[98] S. Yokobori, Y. Watanabe, T. Oshima (**2003**) Mitochondrial genome of Ciona savignyi (Urochordata, Ascidiacea, Enterogona): comparison of gene arrangement and tRNA genes with Halocynthia roretzi mitochondrial genome. *J Mol Evol*, 57(5):574–587.

[99] A. M. Nedelcu, R. W. Lee, C. Lemieux, M. W. Gray, G. Burger (**2000**) The complete mitochondrial DNA sequence of Scenedesmus obliquus reflects an intermediate stage in the evolution of the green algal mitochondrial genome. *Genome Res*, 10(6):819–831.

[100] Y. Hayashi-Ishimaru, T. Ohama, Y. Kawatsu, K. Nakamura, S. Osawa (**1996**) UAG is a sense codon in several chlorophycean mitochondria. *Curr Genet*, 30(1):29–33.

[101] M. J. Laforest, I. Roewer, B. F. Lang (**1997**) Mitochondrial tRNAs in the lower fungus Spizellomyces punctatus: tRNA editing and UAG 'stop' codons recognized as leucine.· . *Nucleic Acids Res*, 25(3):626–632.

[102] S. U. Schneider, M. B. Leible, X. P. Yang (**1989**) Strong homology between the small subunit of ribulose-1,5-bisphosphate carboxylase/oxygenase of two species of Acetabularia and the occurrence of unusual codon usage. *Mol Gen Genet*, 218(3):445–452.

[103] S. U. Schneider, E. J. de Groot (**1991**) Sequences of two rbcS cDNA clones of Batophora oerstedii: structural and evolutionary considerations. *Curr Genet*, 20(1-2):173–175.

[104] A. Liang, K. Heckmann (**1993**) Blepharisma uses UAA as a termination codon. *Naturwissenschaften*, 80(5):225–226.

[105] P. J. Keeling, W. F. Doolittle (**1996**) A non-canonical genetic code in an early diverging eukaryotic lineage. *EMBO J*, 15(9):2285–2290.

[106] A. Kaufmann, M. Knop (**2011**) Genomic promoter replacement cassettes to alter gene expression in the yeast saccharomyces cerevisiae. *Methods Mol Biol*, 765:275–294.

[107] A. A. Brakhage, V. Schroeckh (**2011**) Fungal secondary metabolites - strategies to activate silent gene clusters. *Fungal Genet Biol*, 48(1):15–22.

[108] M. Kanehisa, S. Goto (**2000**) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res*, 28(1):27–30.

[109] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, M. Tanabe (**2012**) KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res*, 40(Database issue):D109–D114.

[110] R. B. Weart, A. H. Lee, A.-C. Chien, D. P. Haeusser, N. S. Hill, P. A. Levin (**2007**) A metabolic sensor governing cell size in bacteria. *Cell*, 130(2):335–347.

[111] G. Krauss (**2008**) *Biochemistry of Signal Transduction and Regulation*. Wiley-VCH, Weinheim, 4 edn.

[112] R. Avraham, Y. Yarden (**2011**) Feedback regulation of EGFR signalling: decision making by early and delayed loops. *Nat Rev Mol Cell Biol*, 12(2):104–117.

[113] A. R. Saltiel, C. R. Kahn (**2001**) Insulin signalling and the regulation of glucose and lipid metabolism. *Nature*, 414(6865):799–806.

[114] L. F. Reichardt (**2006**) Neurotrophin-regulated signalling pathways. *Philos Trans R Soc Lond B Biol Sci*, 361(1473):1545–1564.

[115] J. Andrae, R. Gallini, C. Betsholtz (**2008**) Role of platelet-derived growth factors in physiology and medicine. *Genes Dev*, 22(10):1276–1312.

[116] K. Xie, D. Wei, Q. Shi, S. Huang (**2004**) Constitutive and inducible expression and regulation of vascular endothelial growth factor. *Cytokine Growth Factor Rev*, 15(5):297–324.

[117] C. E. Edling, B. Hallberg (**2007**) c-Kit–a hematopoietic cell essential receptor tyrosine kinase. *Int J Biochem Cell Biol*, 39(11):1995–1998.

[118] R. L. Patterson, D. B. van Rossum, N. Nikolaidis, D. L. Gill, S. H. Snyder (**2005**) Phospholipase C-gamma: diverse roles in receptor-mediated calcium signaling. *Trends Biochem Sci*, 30(12):688–697.

[119] B. D. Manning, L. C. Cantley (**2007**) AKT/PKB signaling: navigating downstream. *Cell*, 129(7):1261–1274.

[120] M. M. McKay, D. K. Morrison (**2007**) Integrating signals from RTKs to ERK/MAPK. *Oncogene*, 26(22):3113–3121.

[121] A. B. Jaffe, A. Hall (**2005**) Rho GTPases: biochemistry and biology. *Annu Rev Cell Dev Biol*, 21:247–269.

[122] D. Chen, M. Zhao, G. R. Mundy (**2004**) Bone morphogenetic proteins. *Growth Factors*, 22(4):233–241.

[123] J. S. Kang, C. Liu, R. Derynck (**2009**) New regulatory mechanisms of TGF-beta receptor function. *Trends Cell Biol*, 19(8):385–394.

[124] W. M. Oldham, H. E. Hamm (**2008**) Heterotrimeric G protein activation by G-protein-coupled receptors. *Nat Rev Mol Cell Biol*, 9(1):60–71.

[125] B. T. MacDonald, K. Tamai, X. He (**2009**) Wnt/beta-catenin signaling: components, mechanisms, and diseases. *Dev Cell*, 17(1):9–26.

[126] D. Pan (**2010**) The hippo signaling pathway in development and cancer. *Dev Cell*, 19(4):491–505.

[127] M. A. Arnaout, S. L. Goodman, J.-P. Xiong (**2002**) Coming to grips with integrin binding to ligands. *Curr Opin Cell Biol*, 14(5):641–651.

[128] L. Matthews, G. Gopinath, M. Gillespie, M. Caudy, D. Croft, B. de Bono, P. Garapati, J. Hemish, H. Hermjakob, B. Jassal, A. Kanapin, S. Lewis, S. Mahajan, B. May, E. Schmidt, I. Vastrik, G. Wu, E. Birney, L. Stein, P. D'Eustachio (**2009**) Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res*, 37(Database issue):D619–D622.

[129] M. Barbieri (**2009**) Three types of semiosis. *Biosemiotics*, 2(1):19–30.

[130] A. Musacchio, E. D. Salmon (**2007**) The spindle-assembly checkpoint in space and time. *Nat Rev Mol Cell Biol*, 8(5):379–393.

[131] B. Ibrahim, S. Diekmann, E. Schmitt, P. Dittrich (**2008**) In-silico modeling of the mitotic spindle assembly checkpoint. *PLoS One*, 3(2):e1555.

[132] E. Szathmáry (**1993**) Coding coenzyme handles: a hypothesis for the origin of the genetic code. *Proc Natl Acad Sci U S A*, 90:9916–9920.

[133] M. Yarus, J. G. Caporaso, R. Knight (**2005**) Origins of the genetic code: the escaped triplet theory. *Annu Rev Biochem*, 74:179–198.

[134] L. Popova-Zeugmann, M. Heiner, I. Koch (**2005**) Time petri net for modelling and analysis of biochemical networks. *Fundamenta Informaticae*, 67:149–162.

[135] M. Heiner, A. Uhrmacher (eds.) (**2011**) *Foundations of formal reconstructionof biochemical networks*, vol. 412 of *J Theoretical Computer Science*.

[136] E. zu Erbach-Schönberg (**2009**) *Simulating the evolution of signalling networks in artificial bacteria*. Diploma thesis, Friedrich-Schiller-Universität Jena.

[137] C. Weisensee (**2011**) *Simulation of the evolution of chemotaxis in virtual cells*. Diploma thesis, Friedrich-Schiller-Universität Jena. In German.

List of Tables

List of Figures

List of Algorithms

Appendix A

Helper methods

A.1 Random network generation

Algorithm A.1 RANDOM(x,y)

Input: Two integers x and y.
Result: A uniformly distributed random number in the range of x and y.

Algorithm A.2 GENERATERANDOMNETWORK()

Input: The size n of the network and the number of reactions m
Result: A random reaction network $N_{rand} = (\mathcal{M}, \mathcal{R})$, with $|\mathcal{M}| = n$ and $|\mathcal{R}| = m$.

1: $\mathcal{M} \leftarrow \bigcup_{i=1}^{m} \{i\}$
2: $\mathcal{R} \leftarrow \emptyset$
3: **for** i in 1 to m **do**
4: $\quad s1 \leftarrow \text{RANDOM}(1, n)$
5: $\quad s2 \leftarrow \text{RANDOM}(1, n)$
6: $\quad s3 \leftarrow \text{RANDOM}(1, n)$
7: $\quad \mathcal{R} \leftarrow \mathcal{R} \cup \{s1 + s2 \rightarrow s3\}$
8: **end for**
9: **return** $N_{rea} = (\mathcal{M}, \mathcal{R})$

For helper method RANDOM see Algorithm A.1 on page 111.

A.2 Methods for the closure-based algorithm

The main algorithm CLOSURECODEFINDER() (Algo. 4.1) is described on page 39.

Algorithm A.3 allClosedSets(A)

Input: A set A of molecular species from network N.
Result: A set B containing all closed sets of A with respect to network N.

```
1: B ← ∅
2: L ← G_CL(A)
3: S ← G_CL({∅})
4: C.add(S) {C is maintained as list}
5: while |C| > 0 do
6:     E ← getFirst(C)
7:     U ← L \ (E ∩ L)
8:     F ← findClosAbove(E, U)
9:     C ← C \ {E}
10:    B ← B ∪ {E}
11:    C ← C ∪ (F \ (F ∩ B))
12: end while
13: return B
```

getFirst returns the first element of a list.

Algorithm A.4 findClosAbove(A,B)

Input: Two sets A, B of molecular species from network N.
Result: A set res of closed sets.

```
1: res ← ∅
2: for all b ∈ B do
3:     B' ← B \ b
4:     A' ← A ∪ b
5:     C ← G_CL(A')
6:     res.add(C)
7: end for
8: return res
```

Algorithm A.5 G_{CL}(A)

Input: An input set $A \subseteq \mathcal{M}$.
Result: A set $B \subseteq \mathcal{M}$ representing the closed set induced by A.

```
1: repeat
2:     B ← A
3:     A ← sqr(B) ∪ B
4: until B == A
5: return B
```

Algorithm A.6 SQR(A,N)

Input: An input set $A \subseteq \mathcal{M}$, with $N = (\mathcal{M}, \mathcal{R})$.
Result: Returns a set $B \subseteq \mathcal{M}$ that can be produced directly by reactions among molecules from A.

```
1: for all ρ ∈ R do
2:     if l_ρ ⊆ A then
3:         B ← result ∪ r_ρ
4:     end if
5: end for
6: return B
```

A.3 Methods for the pathway-based algorithms

The main algorithm PATHCODEFINDER() (Algo. 4.2) is described on page 40.

Algorithm A.7 GETCONTEXT(p,s,t,N)

Input: A reaction path $p = (\rho_1, \rho_2, \ldots, \rho_n)$ from $s \in \mathcal{M}$ to $t \in \mathcal{M}$, with $N = (\mathcal{M}, \mathcal{R})$.
Result: A set $C \subseteq \mathcal{M}$ which is the molecular context of p.

```
1: C ← G_CL({s})
2: for all ρ ∈ p do
3:     C ← C ∪ (l_ρ \ G_CL(C))
4: end for
5: return C
```

Algorithm A.8 GETOUTGOINGREA(A,N)

Input: A reaction network $N = (\mathcal{M}, \mathcal{R})$, and a set $A \in \mathcal{M}$ of molecular species from N.
Result: A random reaction that uses an element of A as reactant.

```
1: cand ← ∅
2: for all ρ ∈ R do
3:     for all s ∈ A do
4:         if s ∈ l_ρ then
5:             cand ← cand ∪ ρ
6:         end if
7:     end for
8: end for
9: r ← RANDOM(1, |cand|)
10: return cand[r]
```

Algorithm A.9 getIncomingRea(A,N)

Input: A reaction network $N = (\mathcal{M}, \mathcal{R})$, and a set $A \in \mathcal{M}$ of molecular species from N.

Result: A random reaction that produces an element of A.

```
1: cand ← ∅
2: for all ρ ∈ R do
3:     for all s ∈ A do
4:         if s ∈ r_ρ then
5:             cand ← cand ∪ ρ
6:         end if
7:     end for
8: end for
9: r ← RANDOM(1, |cand|)
10: return cand[r]
```

Algorithm A.10 getSpecies(R,N)

Input: A set of reactions and a reaction network N.

Result: All species used and produced in the reactions in R.

```
1: S ← ∅
2: for all ρ ∈ R do
3:     S ← S ∪ l_ρ
4:     S ← S ∪ r_ρ
5: end for
6: return S
```

Algorithm A.11 getReactions(R,N)

Input: A set of reactions and a reaction network N.

Result: A set of reactions induced by R.

```
1: R' ← ∅
2: A ← getSpecies(R, N)
3: for all ρ ∈ R do
4:     if l_ρ ∈ A then
5:         R' ← R' ∪ ρ
6:     end if
7: end for
8: return R'
```

Fitting algorithm for random network data

Algorithm A.12 FITMODEL(data,dist)

Input: The random network data *data*. A probability distribution to fit.
Result: A function `model(s,r)` that calculates the model estimate for arbitrary size and density.

for all network sizes s **do**
    $\mu_s \leftarrow$ calculate mean from data of size s.
    $\sigma_s^2 \leftarrow$ calculate variance from data of size s.
    {Identify a suitable scaling factor f}
    $f_s \leftarrow 0$
    for f_s in 1 to 10000 by 0.01 **do**
        if dist==$\mathcal{N}$ **then**
            $\theta_1 = \mu_s$
            $\theta_2 = \sigma_s^2$
        end if
        if dist==ln$\mathcal{N}$ **then**
            $\theta_1 = \log\left(\mu_s / \sqrt{1 + \frac{\sigma_s^2}{\mu_s^2}}\right)$
            $\theta_2 = \log\left(1 + \frac{\sigma_s^2}{\mu_s^2}\right)$
        end if
        if dist==Γ **then**
            $\theta_1 = \frac{\mu_s^2}{\sigma_s^2}$
            $\theta_2 = \frac{\sigma_s^2}{\mu_s}$
        end if
        `fun <- function(r)`{f_s `* dist(r,` θ_1`,` θ_2`) }`
        if `(round(optimize(fun,c(0,200),tol=0.0001,maximum=T,2)==round(maximum,2))` **then**
            break
        end if
    end for
    $means \leftarrow means \cup \mu_s$ {Collect means}
    $variances \leftarrow variances \cup \sigma_s^2$ {Collect variances}
    $factors \leftarrow factors \cup f_s$ {Collect factors}
end for
`fit.mu <- lm(means)` {Fit a linear model of the means over all sizes}
`fit.var <- nls(variances,...)` {Fit a non-linear model of the variances over all sizes}
`fit.factor <- nls(factors,...)` {Fit a non-linear model of the scaling factor over all sizes}
`nullmodel <- function(s,r)`{ {Define function as resulting model}
`m <- fit.mu(s)`
`v <- fit.var(s)`
`f <- fit.factor(s)`
if dist==$\mathcal{N}$ **then**
    $\theta_1 = m$
    $\theta_2 = v$
end if
if dist==ln$\mathcal{N}$ **then**
    $\theta_1 = \log\left(m / \sqrt{1 + \frac{v}{m^2}}\right)$
    $\theta_2 = \log\left(1 + \frac{v}{m^2}\right)$
end if
if dist==Γ **then**
    $\theta_1 = \frac{m^2}{v}$
    $\theta_2 = \frac{v}{m}$
end if
`result <- f * dist(r,`θ_1`,`θ_2`)`
}
return `nullmodel`

The pseudocode contains some functions in R syntax: `lm, optimize, round, nls`. The placeholder dist can be replaced by `dnorm`,`dlnorm` and `dgamma` (package `stats`) depending on the distribution.

Appendix B

Proof of Lemma 3.2.1

We here proof Lemma 3.2.1 from page 28by enumeration.

Lemma 3.2.1 (Ten unique closed sets)*Given an BMC according to Definition 3.2.1 the ten closures* $G_{CL}(s_1), G_{CL}(s_2), G_{CL}(m_1), G_{CL}(m_2), G_{CL}(C), G_{CL}(C'), G_{CL}(s_1 \cup C) = G_{CL}(s_1 \cup C \cup m_1), G_{CL}(s_2 \cup C) = G_{CL}(s_2 \cup C \cup m_2), G_{CL}(s_1 \cup C') = G_{CL}(s_1 \cup C' \cup m_2)$, *and* $G_{CL}(s_2 \cup C') = G_{CL}(s_2 \cup C' \cup m_1)$ *must be different.*

Proof. Given a binary molecular code f we will show the effect of closure equality: If $G_{CL}(s_1) = G_{CL}(s_2)$ then s_1 always leads to the production of s_2 and vice versa, thus the set of signs is degenerated leading to the production of both meanings at the same time, when applying a molecular context. We call this case *sign degeneracy.*
If $G_{CL}(s_1) = G_{CL}(m_1)$ then s_1 always leads to the production of m_1 and vice versa, thus the production of one of the meanings cannot be controlled by the application of a context anymore. The same argument is true for $G_{CL}(s_1) = G_{CL}(m_2)$, $G_{CL}(s_2) = G_{CL}(m_1)$, and $G_{CL}(s_2) = G_{CL}(m_2)$.
If $G_{CL}(m_1) = G_{CL}(m_2)$ then m_1 always leads to the production of m_2 and vice versa, thus the set of meanings is degenerated leading to the production of both meanings at the same time, when applying a molecular context. We call this case *meaning degeneracy.*
If $G_{CL}(s_1) = G_{CL}(C)$ then s_1 always leads to the production of the molecular context C, thus the mapping cannot be controlled any more by this context and one of the meanings then is always present. The same argument is true for $G_{CL}(s_1) = G_{CL}(C')$, $G_{CL}(s_2) = G_{CL}(C)$, $G_{CL}(s_2) = G_{CL}(C')$, $G_{CL}(s_1) = G_{CL}(s_1 \cup C)$, $G_{CL}(s_2) = G_{CL}(s_2 \cup C)$, $G_{CL}(s_1) = G_{CL}(s_1 \cup C')$, $G_{CL}(s_2) = G_{CL}(s_2 \cup C')$.
If $G_{CL}(s_1) = G_{CL}(s_2 \cup C)$ the s_1 alone can generate the context and the other sign, thus this case equivalent to sign degeneracy (fist case). Because $(s_2 \cup C) = G_{CL}(s_2 \cup C \cup m_2)$ s_1 would , in this case also always generate one of the meanings, which destroys the coding property. The same holds for $G_{CL}(s_1) = G_{CL}(s_2 \cup C')$, $G_{CL}(s_2) = G_{CL}(s_1 \cup C)$, and $G_{CL}(s_2) = G_{CL}(s_1 \cup C')$.
If $G_{CL}(m_1) = G_{CL}(C)$ then m_1 produces always the context of its own production, and vice versa C always produces m_1, without any "signalling". The same holds for $G_{CL}(m_1) = G_{CL}(C')$, $G_{CL}(m_2) = G_{CL}(C)$, and $G_{CL}(m_2) = G_{CL}(C')$.
If $G_{CL}(m_1) = G_{CL}(s_1 \cup C)$ then m_1 produces always the context of its own production and the sign, such that m_1 and s_1 would be always present especially also under the alternative context. The same argument holds for $G_{CL}(m_1) = G_{CL}(s_1 \cup C')$, $G_{CL}(m_2) = G_{CL}(s_1 \cup C)$, $G_{CL}(m_2) = G_{CL}(s_1 \cup C')$, $G_{CL}(m_1) = G_{CL}(s_2 \cup C)$, $G_{CL}(m_1) = G_{CL}(s_2 \cup C')$, $G_{CL}(m_2) = G_{CL}(s_2 \cup C)$, and $G_{CL}(m_2) = G_{CL}(s_2 \cup C')$.

If $G_{CL}(C) = G_{CL}(C')$ then both contexts are always present and no distinguishable mapping can be established. The same argument is true for $G_{CL}(C) = G_{CL}(s_1 \cup C')$, $G_{CL}(C) = G_{CL}(s_2 \cup C')$, $G_{CL}(C') = G_{CL}(s_1 \cup C)$, and $G_{CL}(C') = G_{CL}(s_2 \cup C)$. We call this case *context degeneracy*.
If $G_{CL}(C) = G_{CL}(s_1 \cup C)$ then the context C alone produces the sign and always triggers the production of m_1 which is against the coding property. The same is argument holds for $G_{CL}(C) = G_{CL}(s_2 \cup C)$, $G_{CL}(C') = G_{CL}(s_1 \cup C')$, and $G_{CL}(C') = G_{CL}(s_2 \cup C')$.
The cases $G_{CL}(s1 \cup C) = G_{CL}(s_2 \cup C)$, $G_{CL}(s1 \cup C') = G_{CL}(s_2 \cup C')$ are a form of sign degeneracy.
The cases $G_{CL}(s1 \cup C) = G_{CL}(s_1 \cup C')$, $G_{CL}(s2 \cup C) = G_{CL}(s_2 \cup C')$ are a form of context degeneracy.
The cases $G_{CL}(s1 \cup C) = G_{CL}(s_2 \cup C')$, $G_{CL}(s2 \cup C) = G_{CL}(s_1 \cup C')$ are a mixed form of context and sign degeneracy.
In conclusion we see that all 45 combinations of these ten closed sets lead to some problem with the code conditions and thus they have to be different to establish a binary molecular code. □

Appendix C

Potential codes in signal transduction

Table C.1 Summary of the components of the codes found in the human signal transduction network.

#	molecular species	#codes with semiotic role: sign	meaning	context	biological role
1	14-3-3proteinbeta/alphacytosol	0	1	0	COF
2	3-5-CyclicAMPcytosol	0	0	5	SM
3	ActivatedFGFRFRS2alphaplasmamembraneextracellularregion	0	0	16	AR
4	ActivatedFGFRp-FRS2alphaGRB2GAB1PI3Kplasmamembranecytosolextracellularregion	0	7	0	AR
5	ActivatedFGFRp-FRS2alphaGRB2GAB1PI3KR1plasmamembranecytosolextracellularregion	0	9	0	AR
6	ActivatedFGFRp-FRS2alphaplasmamembrane	0	0	16	AR
7	ActivatedFGFRp-SHC1GRB2SOS1plasmamembranecytosolextracellularregion	0	10	0	AR
8	ActivatedFGFRp-SHC1plasmamembrane	0	0	10	AR
9	ActivatedFGFRSHC1plasmamembraneextracellularregioncytosol	0	0	10	AR
10	ActivatedmTORC1cytosol	0	1	0	ST
11	ActivatedPLCbeta1/4plasmamembrane	18	0	105	ST
12	ActivatedRac1PI3Kalphacytosol	0	1	0	ST
13	ActivatedROCKRhoA/B/CGTPplasmamembrane	0	2	0	ST
14	ActivatedTrkAreceptorcomplexClathrin-coatedvesicleplasmamembraneextracellularregion	0	0	9	AR
15	Activec-SRCp-Y419plasmamembrane	0	1	0	AR
16	ActiveTrkAreceptorcomplexRIT/RIN-GDPplasmamembrane	0	0	4	AR
17	ActiveTrkreceptorcomplexRIT/RIN-GTPB-RAFplasmamembrane	0	1	0	AR
18	ActiveTrkreceptorcomplexRIT/RIN-GTPplasmamembrane	0	2	0	AR
19	AdenylatecyclaseMg2plusCOFplasmamembrane	0	0	0	ST
20	ADPcytosol	0	3	0	COF
21	AMPcytosol	0	0	5	COF
22	AMPKheterotrimercytosol	0	0	5	COF
23	AP-2complexplasmamembrane	0	0	1	COF
24	ATPcytosol	110	0	3	AR
25	B-RAFcytosol	0	0	1	COF
26	beta-NGFdimerTrkAreceptordimerplasmamembrane	0	0	2	LR
27	CBLcytosol	0	0	33	COF
28	CBLGRB2cytosol	43	0	19	COF
29	CBLplasmamembrane	0	0	4	COF
30	CIN85Endophilincytosol	0	0	2	COF
31	Clathrinplasmamembrane	0	0	1	COF
32	EGFEGFRdimerplasmamembraneextracellularregion	0	0	29	LR
33	EGFp-6Y-EGFRCBLBeta-PixCDC42GTPplasmamembrane	0	0	4	LR
34	EGFp-6Y-EGFRCBLGRB2plasmamembrane	0	20	2	LR
35	EGFp-6Y-EGFRCBLp-Y53/55-SPRY1/2plasmamembrane	0	0	1	LR
36	EGFp-6Y-EGFRCBLplasmamembraneextracellularregion	0	4	3	LR
37	EGFp-6Y-EGFRCBLUb-p-Y53/55-SPRY1/2plasmamembrane	0	1	0	LR
38	EGFp-6Y-EGFRGRB2GAB1PIK3cytosolextracellularregionplasmamembrane	0	1	0	LR
39	EGFp-6Y-EGFRGRB2GAB1PIK3R1plasmamembranecytosolextracellularregion	0	1	10	LR
40	EGFp-6Y-EGFRGRB2GAB1plasmamembranecytosolextracellularregion	0	18	0	LR

Summary of the components of the codes found in the signal transduction network (cont.)

#	molecular species	#codes with semiotic role: sign	meaning	context	biological role
41	EGFp-6Y-EGFRGRB2SOS1plasmamembranecytosolextracellularregion	0	10	0	LR
42	EGFp-6Y-EGFRp-Y349350-SHC1GRB2SOS1plasmamembrane	0	8	0	LR
43	EGFp-6Y-EGFRp-Y349350-SHC1plasmamembrane	0	0	8	LR
44	EGFp-6Y-EGFRp-Y371-CBLCIN85EndophilinEpsinEps15REps15plasmamembrane	0	0	1	LR
45	EGFp-6Y-EGFRp-Y371-CBLGRB2CIN85EndophilinEpsinEps15REps15Clathrinplasmamembrane	0	1	0	LR
46	EGFp-6Y-EGFRp-Y371-CBLGRB2CIN85Endophilinplasmamembrane	0	0	10	LR
47	EGFp-6Y-EGFRp-Y371-CBLGRB2plasmamembrane	0	0	1	LR
48	EGFp-6Y-EGFRp-Y371-CBLplasmamembrane	0	0	5	LR
49	EGFp-6Y-EGFRp-Y371-CBLUb-CIN85EndophilinEpsinEps15REps15plasmamembrane	0	1	0	LR
50	EGFp-6Y-EGFRplasmamembraneextracellularregion	1	0	51	LR
51	EGFp-6Y-EGFRPLCG1plasmamembraneextracellularregioncytosol	0	0	18	LR
52	EGFp-6Y-EGFRSHC1plasmamembranecytosolextracellularregion	0	0	8	LR
53	EGFp-EGFRp-ERBB2GRB2GAB1PI3Kplasmamembranecytosolextracellularregion	0	5	0	LR
54	EGFp-EGFRp-ERBB2GRB2GAB1PIK3R1plasmamembranecytosolextracellularregion	0	5	5	LR
55	EGFp-EGFRp-ERBB2GRB2GAB1plasmamembranecytosolextracellularregion	0	13	0	LR
56	EGFp-EGFRp-ERBB2GRB2SOS1plasmamembranecytosolextracellularregion	0	13	0	LR
57	EGFp-EGFRp-ERBB2PLCG1plasmamembranecytosolextracellularregion	0	0	36	LR
58	EGFUb-p-6Y-EGFRp-Y371-CBLGRB2plasmamembrane	0	1	0	LR
59	EGFUb-p-6Y-EGFRp-Y371-CBLplasmamembrane	0	1	4	LR
60	Endophilinplasmamembrane	0	0	9	COF
61	Eps15HGSSTAMcytosol	0	0	2	EFF
62	Eps15Rplasmamembrane	0	0	2	EFF
63	Epsinplasmamembrane	0	0	2	EFF
64	ERBB3plasmamembrane	0	0	1	R
65	ERBB3RNF41cytosolplasmamembrane	0	2	2	EFF
66	G-alphat-GDPG-beta-gammaplasmamembrane	0	0	26	ST
67	G-alphat-GTPplasmamembrane	0	1	0	ST
68	G-proteinalpha12/13GDPplasmamembrane	0	1	0	ST
69	G-proteinalpha12/13GTPplasmamembrane	0	1	0	ST
70	G-proteinalpha12/13LARGplasmamembrane	0	1	0	ST
71	G-proteinalpha12/13LARGPlexinB1plasmamembrane	0	1	0	ST
72	G-proteinalphaGDPplasmamembrane	0	14	102	ST
73	G-proteinalphaiGDPplasmamembrane	0	1	307	ST
74	G-proteinalphaiGTPAdenylatecyclaseplasmamembrane	70	1	56	ST
75	G-proteinalphaiGTPplasmamembrane	0	22	0	ST
76	G-proteinalphaq/11GDPplasmamembrane	0	0	1	ST
77	G-proteinalphaq/11GTPplasmamembrane	0	0	95	ST
78	G-proteinalphasGDPplasmamembrane	0	1	4	ST
79	G-proteinalphasGTPplasmamembrane	0	1	153	ST
80	G-proteinalphazGDPplasmamembrane	0	1	5	ST
81	G-proteinalphazGTPAdenylatecyclaseplasmamembrane	0	152	0	ST
82	G-proteinalphazGTPplasmamembrane	0	1	151	ST
83	G-proteinbeta-gammacomplexplasmamembrane	1	0	429	ST
84	GAB1cytosol	2	0	62	COF

Summary of the components of the codes found in the signal transduction network (cont.)

#	molecular species	#codes with semiotic role: sign	meaning	context	biological role
85	Galpha-olfGTPplasmamembrane	56	0	89	ST
86	GDPcytosol	15	17	34	COF
87	Gialpha1GDPAdenylatecyclaseGalpha-olfGDPplasmamembrane	113	0	0	ST
88	Gialpha1GTPAdenylatecyclaseGalpha-olfGTPplasmamembrane	140	0	89	ST
89	GPCRligandcomplexesthatactonGsHeterotrimericG-proteinGsactiveplasmamembrane	46	1	0	AR
90	GPCRsthatactivateG12/13plasmamembrane	0	1	0	R
91	GPCRsthatactivateGiplasmamembrane	0	22	0	R
92	GPCRsthatactivateGsplasmamembrane	0	1	0	R
93	GPCRsthatactivateGzplasmamembrane	0	1	0	R
94	Gprotein-GDPcomplexplasmamembrane	18	29	57	ST
95	GproteinalphaGTPcomplexplasmamembrane	18	2	201	ST
96	GRB2boundtopFADK1inFocaladhesionplasmamembrane	0	3	0	ST
97	GRB2cytosol	43	0	9	COF
98	GRB2GAB1cytosol	4	0	62	COF
99	GRB2GAB1PIK3R1cytosol	0	0	18	ST
100	GRB2GAB1PIP3plasmamembranecytosol	0	15	13	ST
101	GRB2p-SHP2p-KITcomplexplasmamembranecytosolextracellularregion	0	3	0	R
102	GRB2SOS1cytosol	0	0	62	ST
103	GRB2SOS1p-KITcomplexplasmamembranecytosolextracellularregion	0	5	0	R
104	GRB2SOS1p-Y349350-SHC1p-ERBB4plasmamembranecytosolextracellularregion	0	6	0	R
105	GRB2SOS1p-Y349350-SHC1PhosphorylatedERBB2heterodimersplasmamembranecytosolextracellularregion	0	10	0	AR
106	Gs-activatedadenylatecyclaseplasmamembrane	0	154	0	ST
107	GTPcytosol	36	1	298	COF
108	H2Ocytosol	111	0	24	COF
109	HeterotrimericG-proteinGiinactiveplasmamembrane	0	114	0	ST
110	HeterotrimericG-proteinGq/11inactiveplasmamembrane	6	0	6	ST
111	HeterotrimericG-proteinGsinactiveplasmamembrane	14	0	14	ST
112	HeterotrimericG-proteinGzinactiveplasmamembrane	9	0	9	ST
113	IkBalphaNF-kBcomplexcytosol	0	0	1	EFF
114	IntegrinalphaIIbbeta3pY530-SRCCSKplasmamembrane	0	0	10	R
115	IntegrinalphaIIbbeta3pY530-SRCCSKTalinRIAMcomplexplasmamembrane	0	14	3	R
116	IntegrinalphaIIbbeta3pY530-SRCCSKTalinRIAMcomplexpY317-SHCplasmamembrane	0	1	0	R
117	IntegrinalphaIIbbeta3pY530-SRCCSKTalinRIAMcomplexSHCplasmamembrane	0	1	0	R
118	KITsSCFdimerKITplasmamembraneextracellularregion	0	0	5	R
119	Largvariant1cytosol	0	0	2	ST
120	LigandGPCRcomplexesthatactivateG12/13HeterotrimericG-proteinG12/13active.plasmamembrane	1	1	0	LR
121	LigandGPCRcomplexesthatactivateG12/13HeterotrimericG-proteinG12/13inactive.plasmamembrane	0	0	8	LR
122	LigandGPCRcomplexesthatactivateG12/13plasmamembrane	0	0	1	LR

Summary of the components of the codes found in the signal transduction network (cont.)

#	molecular species	#codes with semiotic role: sign	meaning	context	biological role
123	LigandGPCRcomplexesthatactivateGiHeterotrimericG-proteinGiactiveplasmamembrane	2	22	0	LR
124	LigandGPCRcomplexesthatactivateGiHeterotrimericG-proteinGiinactiveplasmamembrane	0	204	6	LR
125	LigandGPCRcomplexesthatactivateGiplasmamembrane	0	0	214	LR
126	LigandGPCRcomplexesthatactivateGq/11HeterotrimericG-proteinGqactiveplasmamembrane	62	0	0	LR
127	LigandGPCRcomplexesthatactivateGq/11HeterotrimericG-proteinGqinactiveplasmamembrane	6	0	18	LR
128	LigandGPCRcomplexesthatactivateGq/11plasmamembrane	6	0	8	LR
129	LigandGPCRcomplexesthatactivateGsHeterotrimericG-proteinGsinactiveplasmamembrane	33	0	84	LR
130	LigandGPCRcomplexesthatactivateGsplasmamembrane	14	0	19	LR
131	LigandGPCRcomplexesthatactivateGzHeterotrimericG-proteinGzactiveplasmamembrane	40	1	0	LR
132	LigandGPCRcomplexesthatactivateGzHeterotrimericG-proteinGzinactiveplasmamembrane	33	0	87	LR
133	LigandGPCRcomplexesthatactivateGzplasmamembrane	9	0	15	LR
134	LigandsofGPCRsthatactivateG12/13extracellularregion	0	1	0	L
135	LigandsofGPCRsthatactivateGiextracellularregion	0	22	0	L
136	LigandsofGPCRsthatactivateGsextracellularregion	0	1	0	L
137	LigandsofGPCRsthatactivateGzextracellularregion	0	1	0	L
138	mLst8cytosol	0	0	1	ST
139	mTORcytosol	0	0	1	ST
140	Mu-typeopioidreceptorplasmamembrane	18	0	18	R
141	NGFligandp75NTRIRAK1MYD88plasmamembrane	0	0	1	LR
142	NGFligandp75NTRPhospho-IRAK1polyubiquitinatedTRAF6p62plasmamembrane	0	1	0	LR
143	NGFligandp75NTRPhospho-IRAK1TRAF6p62plasmamembrane	0	0	1	LR
144	NRG1/2p-10Y-ERBB3p-ERBB2RNF41cytosolplasmamembraneextracellularregion	0	2	2	AR
145	OpioidMORG-proteincomplexplasmamembrane	1	20	13	LR
146	OpioidMORGprotein-GDPcomplexplasmamembrane	36	51	54	LR
147	OpioidMORGprotein-GTPcomplexplasmamembrane	17	2	0	LR
148	OpioidMORplasmamembrane	18	0	118	LR
149	Opioidpeptideextracellularregion	18	0	18	L
150	p-7Y-KITsSCFdimerp-7Y-KITplasmamembraneextracellularregion	0	0	5	LR
151	p-AMPKheterotrimerAMPcytosol	0	5	0	LR
152	p-AMPKheterotrimercytosol	0	0	5	R
153	p-Raf1S259S62114-3-3proteinbeta/alphacytosol	0	1	4	ST
154	p-S32S36-IkBAcytosol	0	0	1	ST
155	p-S33S37T41S45-beta-cateninAxinCK1alphaGSK3Bphospho-APC20aarepeatregionPP2AFAM123Bcomplexcytosol	0	0	15	ST
156	p-S33S37T41S45-beta-cateninAxinGSK3CK1alphaAPCPP2AFAM123Bcomplexcytosol	0	0	15	ST
157	p-SHP2p-KITcomplexplasmamembranecytosolextracellularregion	0	0	3	R
158	p-USP8cytosol	0	0	16	EFF
159	p-Y349350-SHC1p-ERBB4plasmamembranecytosolextracellularregion	0	0	6	R
160	p-Y349350-SHC1PhosphorylatedERBB2heterodimersplasmamembranecytosolextracellularregion	0	0	10	R
161	p21RASGDPplasmamembranecytosol	0	0	8	ST
162	p21RASGTPplasmamembranecytosol	0	1	0	ST
163	p62cytosol	0	0	1	ST
164	PDGFPhospho-PDGFreceptordimerplasmamembrane	0	0	1	LR
165	PhosphorylatedERBB2EGFRheterodimersplasmamembraneextracellularregion	0	0	36	AR
166	PhosphorylatedERBB2ERBB3heterodimersplasmamembraneextracellularregion	0	0	1	AR

Summary of the components of the codes found in the signal transduction network (cont.)

#	molecular species	#codes with semiotic role: sign	meaning	context	biological role
167	PI345P3plasmamembrane	0	1	72	ST
168	PI3Kalphacytosol	0	0	1	ST
169	PI45P2plasmamembrane	0	0	57	ST
170	Picytosol	0	21	0	COF
171	PIK3CAcytosol	0	0	13	ST
172	PIK3R1cytosol	0	0	18	ST
173	PIK3R1plasmamembrane	0	0	10	ST
174	PLC-betacytosol	0	0	99	ST
175	PLCbetaGalphaq/11plasmamembrane	0	95	0	ST
176	PlexinB1plasmamembrane	0	0	1	R
177	PP2AACcytosol	0	6	0	ST
178	PP2AACSPRY2plasmamembrane	0	0	5	ST
179	PP2AACY55/Y227-pSPRY2plasmamembrane	0	0	6	ST
180	PPA2AACY55/Y227p-SPRY2GRB2plasmamembrane	39	0	0	ST
181	pS27-GproteinalphazGTPplasmamembrane	0	1	0	ST
182	pY317-SHCcytosol	0	1	0	ST
183	RAC1-GDPcytosol	15	17	35	ST
184	RAC1-GTPcytosol	36	2	298	ST
185	RACGDPplasmamembrane	0	0	1	ST
186	RACGTPplasmamembrane	0	1	0	ST
187	RAL-GDPcytosol	15	17	36	ST
188	RAL-GTPcytosol	36	2	298	ST
189	RalGDScytosol	0	0	1	ST
190	Rap1-GDPcytosol	15	17	35	ST
191	Rap1-GDPplasmamembrane	12	0	22	ST
192	Rap1-GTPcytosol	36	2	298	ST
193	Rap1-GTPPIP2RIAMplasmamembrane	0	19	0	ST
194	Rap1-GTPplasmamembrane	0	1	36	ST
195	Raptorcytosol	0	0	1	ST
196	Ras-GTPRalGDScomplexplasmamembranecytosol	0	1	0	ST
197	RASRAF14-3-3plasmamembrane	0	1	0	ST
198	RASRAFplasmamembrane	0	1	0	ST
199	RhebGDPcytosol	15	17	41	ST
200	RhebGTPcytosol	36	2	298	ST
201	RhoA/B/CGTPplasmamembrane	0	1	0	ST
202	RhoABCGDPplasmamembrane	0	0	2	ST
203	RhoGTPaseGDPplasmamembrane	0	0	14	ST
204	RhoGTPaseGTPplasmamembrane	0	1	0	ST

Summary of the components of the codes found in the signal transduction network (cont.)

#	molecular species	#codes semiotic role			biological role
		sign	meaning	context	
205	RIAMcytosol	0	0	36	ST
206	RIT/RIN-GDPplasmamembrane	0	0	2	ST
207	RNF41cytosol	0	1	1	EFF
208	ROCKcytosol	0	0	1	ST
209	SCF-beta-TrCP1complexassociatedwithphosphorylatedbeta-catenincytosol	0	15	0	LR
210	SCF-beta-TrCP1complexcytosol	0	0	15	LR
211	SHC1cytosol	1	0	3	ST
212	SHC1p-ERBB4plasmamembranecytosolextracellularregion	0	0	6	R
213	SHC1PhosphorylatedERBB2heterodimersplasmamembranecytosolextracellularregion	0	0	10	R
214	SHCactivatedinsulinreceptorplasmamembrane	0	0	3	AR
215	SHP2SFKsp-KITsSCFdimerp-KITplasmamembranecytosolextracellularregion	0	0	3	AR
216	SOS1cytosol	0	0	62	ST
217	SPRY1/2cytosol	0	0	1	ST
218	SRCplasmamembrane	1	0	1	ST
219	Talin-1cytosol	0	0	17	ST
220	TalinRIAMcomplexECMligandsalphaIIbbeta3Activep-Y419-SRCpY397-FADK1plasmamembrane	0	0	3	LR
221	TalinRIAMcomplexECMligandsIntegrinalphaIIbbeta3Activep-Y419-SRCpY397407576577861925-FADK1plasmamembrane	0	0	3	LR
222	TalinRIAMcomplexplasmamembrane	0	7	0	ST
223	TRAF6cytosol	0	0	1	ST
224	Ub-RNF41cytosol	0	1	15	EFF
225	Ub-RNF41p-USP8cytosol	0	1	15	EFF
226	Ub-Y55/Y227p-SPRY2cytosolplasmamembrane	0	5	0	ST
227	ubiquitinatedphospho-beta-cateninSCFbeta-TrCP1complexcytosol	16	0	0	R
228	Ubiquitinatedphospho-IkBcytosol	0	1	0	R
229	ubiquitincytosol	1	0	16	COF
230	UbY55/Y227-pSPRY2CBLplasmamembrane	0	5	14	ST
231	USP8cytosol	0	0	16	EFF
232	VAV1Rho/RacEFFsGDPcytosol	15	17	34	EFF
233	VAV1Rho/RacEFFsGTPcytosol	36	1	298	EFF
234	Y55/Y227-pSPRY2CBLplasmamembrane	0	0	19	ST

Appendix D

Potential codes in metabolism

Table D.1 Summary of the components of the codes found in the KEGG metabolic networks.

#	Domain		Codomain		Molecular contexts	
1	C00527	C00007	C04480	C05116	C07282, C00028, C00030, C00877, C00682,	C00028, C00011, C00001, C00090,
2	C02411	C00007	C04480	C05116	C07282, C00030, C00877, C00682,	C00011, C00001, C00090,
3	C00007	C00026	C04480	C00302	C00086, C00014, C00090,	C00682, C05715, C00090,
4	C00007	C00026	C04480	C00302	C00011, C00014, C00177, C00090,	C00232, C00682, C06059, C00177, C00090,
5	C04522	C00007	C04480	C07091	C07090, C00682, C00090,	C00026, C00001, C06659,
6	C00007	C00026	C04480	C05715	C00011, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
7	C00007	C00026	C04480	C05715	C00011, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
8	C00007	C00026	C04480	C00302	C00011, C06059, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
9	C00007	C00026	C04480	C00302	C00011, C06059, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
10	C00028	C00007	C11936	C04480	C00011, C11934, C00001, C00090,	C00527, C00682, C11935, C00090,
11	C00027	C00026	C04480	C00302	C04905, C00011, C00014,	C00682, C05715, C00090,
12	C03585	C00007	C03676	C04480	C00026, C00001, C06659,	C00682, C04431, C00090,
13	C04522	C00007	C02222	C04480	C00026, C00001, C06659,	C00682, C04431, C00090,
14	C03585	C00007	C02222	C04480	C00026, C00001, C06659,	C00682, C04431, C00090,
15	C00007	C00026	C04480	C00302	C00011, C06059, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
16	C00007	C00026	C04480	C00302	C04905, C00011, C00014, C00090,	C05636, C00682, C05715,
17	C04522	C00007	C03676	C04480	C00026, C00001, C06659,	C00682, C04431, C00090,
18	C03585	C00007	C07091	C04480	C00011, C00001, C00090,	C07090, C00682, C00090,
19	C00028	C00007	C04480	C05116	C00527, C00877, C00682, C00090,	C00527, C00011, C00001, C00090,
20	C00793	C00026	C06201	C05829	C00334, C00232, C00001, C00014,	C00022, C00025, C00001,
21	C05636	C00026	C05715	C00237	C00232, C00007, C00177, C00027,	C00007, C00011, C00014,
22	C00007	C00026	C04480	C05715	C00011, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
23	C00007	C00026	C04480	C05715	C00011, C06059, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
24	C00007	C00026	C04480	C05715	C00011, C00014, C00090,	C00232, C04905, C00682, C00177, C00090,
25	C00007	C00026	C04480	C00302	C00011, C06059, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
26	C03585	C00007	C04480	C07091	C07090, C00682, C00090,	C00026, C00001, C06659,
27	C00007	C00026	C04480	C05715	C00011, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
28	C00026	C00097	C06201	C05829	C00022, C00025, C00001,	C00334, C00232, C00001, C00014,
29	C00007	C00026	C04480	C00302	C00011, C00014, C00177, C00090,	C00232, C00682, C06059, C00177, C00090,
30	C03585	C00007	C02222	C04480	C00011, C00001, C00090,	C00682, C04431, C00090,
31	C05636	C00026	C00302	C00237	C00232, C00007, C00177, C00027,	C04905, C00007, C00011, C00014,
32	C00027	C00026	C04480	C05715	C00011, C00014,	C05636, C00232, C00682, C00177,
33	C03585	C00007	C03676	C04480	C00011, C00001, C00090,	C00682, C04431, C00090,
34	C00007	C00026	C04480	C05715	C00011, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
35	C00007	C03453	C04480	C03589	C00011, C00001, C00090,	C00596, C00682, C00090,
36	C00007	C00026	C04480	C05715	C00011, C06059, C00014, C00090,	C00232, C00682, C06059, C00177, C00090,
37	C00007	C00026	C04480	C00302	C00011, C00014, C00177, C00090,	C00232, C00682, C06059, C00177, C00090,

Molecular species are given in KEGG compound id's. For the list of species see Table D.2

Table D.2 Summary of the components of the codes found in the KEGG metabolic networks.

#	KEGG ID	Compound name
1	cpd:C00001	H2O; Water
2	cpd:C00007	Oxygen; O2
3	cpd:C00011	CO2; Carbon dioxide
4	cpd:C00014	NH3; Ammonia
5	cpd:C00022	Pyruvate; Pyruvic acid; 2-Oxopropanoate;
6	cpd:C00025	L-Glutamate; L-Glutamic acid; Glutamate
7	cpd:C00026	2-Oxoglutarate; Oxoglutaric acid; alpha-Ketoglutaric acid
8	cpd:C00027	Hydrogen peroxide; H2O2
9	cpd:C00028	Acceptor; Hydrogen-acceptor; A; Oxidized donor
10	cpd:C00030	Reduced acceptor; AH2; Hydrogen-donor; Donor
11	cpd:C00086	Urea; Carbamide
12	cpd:C00090	Catechol; 1,2-Benzenediol; o-Benzenediol
13	cpd:C00097	L-Cysteine; L-2-Amino-3-mercaptopropionic acid
14	cpd:C00177	Cyanide; Prussiate; CN-; Cyano
15	cpd:C00232	Succinate semialdehyde; Succinic semialdehyde; 4-Oxobutanoate
16	cpd:C00237	CO; Carbon monoxide
17	cpd:C00302	DL-Glutamate; Glutamate; Glutamic acid
18	cpd:C00334	4-Aminobutanoate; 4-Aminobutanoic acid; 4-Aminobutyrate; GABA
19	cpd:C00527	Glutaryl-CoA
20	cpd:C00596	2-Hydroxy-2,4-pentadienoate; cis-2-Hydroxypenta-2,4-dienoate;
21	cpd:C00682	2-Hydroxymuconate semialdehyde; 2-Hydroxymuconic semialdehyde
22	cpd:C00793	D-Cysteine; D-Amino-3-mercaptopropionic acid
23	cpd:C00877	Crotonoyl-CoA; Crotonyl-CoA; 2-Butenoyl-CoA; trans-But-2-enoyl-CoA
24	cpd:C02222	2-Maleylacetate; 4-Oxohex-2-enedioate
25	cpd:C02411	Glutaconyl-1-CoA; 4-Carboxybut-2-enoyl-CoA
26	cpd:C03453	gamma-Oxalocrotonate; (Z)-5-Oxohex-2-enedioate; 4-Oxalocrotonate
27	cpd:C03585	3-Chloro-cis,cis-muconate
28	cpd:C03589	4-Hydroxy-2-oxopentanoate; 4-Hydroxy-2-oxovalerate
29	cpd:C03676	3-Hydroxy-cis,cis-muconate
30	cpd:C04431	cis-4-Carboxymethylenebut-2-en-4-olide; 4-Carboxymethylenebut-2-en-4-olide
31	cpd:C04480	3-Carboxy-2-hydroxymuconate semialdehyde
32	cpd:C04522	2-Chloro-2,5-dihydro-5-oxofuran-2-acetate; 5-Chloro-2,5-dihydro-2-oxofuran-5-acetate
33	cpd:C04905	1-(4-Amino-2-methylpyrimid-5-ylmethyl)-3-(beta-hydroxyethyl)-2-methylpyridinium bromide
34	cpd:C05116	3-Hydroxybutanoyl-CoA
35	cpd:C05636	3-Hydroxykynurenamine
36	cpd:C05715	gamma-Amino-gamma-cyanobutanoate; 4-Amino-4-cyanobutanoic acid
37	cpd:C05829	N-Carbamyl-L-glutamate
38	cpd:C06059	Cyclic amidines
39	cpd:C06201	2,4-Dihydroxyhept-2-enedioate; 2,4-Dihydroxyhept-2-1,7-dioate
40	cpd:C06659	Dihydroclavaminic acid; Dihydroclavaminate
41	cpd:C07090	Protoanemonin; 4-Methylenebut-2-en-4-olide; cis-4-Methylenebut-2-en-4-olide
42	cpd:C07091	cis-Acetylacrylate
43	cpd:C07282	[eIF5A-precursor]-deoxyhypusine; Protein N6-(4-Aminobutyl)-L-lysine
44	cpd:C11934	2-Hydroxy-4-isopropenylcyclohexane-1-carboxyl-CoA
45	cpd:C11935	4-Isopropenyl-2-oxy-cyclohexanecarboxyl-CoA; 4-Isopropenyl-2-ketocyclohexane-1-carboxyl-CoA
46	cpd:C11936	3-Isopropenylpimelyl-CoA

Appendix E

Networks

The network models are all in REA-format. The REA-format is a plain text format for chemical reaction networks and basically contains the number of molecular species, a list of molecular species, the number of reactions and the list of reactions. Stoichiometric information is maintained, while kinetics are not represented in .rea-files.
All networks are provided on the supplementary CD.

E 1 Example networks

E 1.1 BMC 1

```
# Number of Components
8
# Components
A1
A2
B1
B2
E1
E2
E3
E4
# Number of Reactions
4
# Reactions
1 A1 1 E1 -> 1 B1 1 E1
1 A1 1 E2 -> 1 B2 1 E2
1 A2 1 E3 -> 1 B1 1 E3
1 A2 1 E4 -> 1 B2 1 E4
```

E 1.2 BMC 2

```
# Number of Components
6
# Components
A1
A2
B1
B2
E1
E2
# Number of Reactions
4
# Reactions
1 A1 1 E1 -> 1 B1 1 E1
1 A1 1 E2 -> 1 B2 1 E2
1 A2 1 E2 -> 1 B1 1 E2
1 A2 1 E1 -> 1 B2 1 E1

```

E 1.3 Extended BMC

```
# reactions genetic code made by hand
# number of molecules:
16
# molecules:
m1
m2
m3
m4
m5
m6
m7
m8
e1
e2
e3
e4
e5
e6
e7
e8
# number of rules:
8
# rules:
1 m1 1 e1 -> 1 m3
1 m1 1 e2 -> 1 m4
1 m2 1 e3 -> 1 m3
1 m2 1 e4 -> 1 m4
1 m5 1 e5 -> 1 m1
1 m6 1 e6 -> 1 m2
1 m3 1 e7 -> 1 m7
1 m4 1 e8 -> 1 m8
```

E 2 Combustion chemistries

E 2.1 Dimethyl ether

```
# Number of Components:
79
# Components:
h
h2
o
o2
oh
h2o
n2
co
hco
co2
ch3
ch4
ho2
h2o2
ch2o
ch3o
c2h6
c2h4
c2h5
ch2
ch
c2h
c2h2
c2h3
ch3oh
ch2oh
ch2co
hcco
c2h5oh
pc2h4oh
sc2h4oh
ch3co
ch2cho
ch3cho
ch3coch3
ch3coch2
c2h5cho
c2h5co
```

```
c2h5o
ch3o2
c2h5o2
ch3o2h
c2h5o2h
c2h3o1-2
ch3co2
c2h4o1-2
c2h4o2h
o2c2h4oh
ch3co3
ch3co3h
c2h3co
c2h3cho
ch3coch2o2
ch3coch2o2h
ch3coch2o
hco3h
hco3
hco2
o2c2h4o2h
ch2(s)
ch3och3
ch3och2
ch3och2o2
ch2och2o2h
ch3och2o2h
ch3och2o
o2ch2och2o2h
ho2ch2ocho
och2ocho
hoch2oco
hoch2o
hco2h
ch3ocho
ch3oco
ch2ocho
ch3och2oh
hoch2o2h
och2o2h
hoch2o2
# Number of Reactions:
708
# Reactions:
1 ch3 1 h -> 1 ch4
1 ch4 -> 1 ch3 1 h
1 ch4 1 h -> 1 ch3 1 h2
1 ch3 1 h2 -> 1 ch4 1 h
1 ch4 1 oh -> 1 ch3 1 h2o
1 ch3 1 h2o -> 1 ch4 1 oh
1 ch4 1 o -> 1 ch3 1 oh
1 ch3 1 oh -> 1 ch4 1 o
1 c2h6 1 ch3 -> 1 c2h5 1 ch4
1 c2h5 1 ch4 -> 1 c2h6 1 ch3
1 hco 1 oh -> 1 co 1 h2o
1 co 1 h2o -> 1 hco 1 oh
1 co 1 oh -> 1 co2 1 h
1 co2 1 h -> 1 co 1 oh
1 h 1 o2 -> 1 o 1 oh
1 o 1 oh -> 1 h 1 o2
1 o 1 h2 -> 1 h 1 oh
1 h 1 oh -> 1 o 1 h2
1 o 1 h2o -> 1 oh 1 oh
1 oh 1 oh -> 1 o 1 h2o
1 oh 1 h2 -> 1 h 1 h2o
1 h 1 h2o -> 1 oh 1 h2
1 hco -> 1 h 1 co
1 h 1 co -> 1 hco
1 h2o2 1 oh -> 1 h2o 1 ho2
1 h2o 1 ho2 -> 1 h2o2 1 oh
1 c2h4 1 o -> 1 ch3 1 hco
1 ch3 1 hco -> 1 c2h4 1 o
1 h 1 c2h4 -> 1 c2h5
1 c2h5 -> 1 h 1 c2h4
1 ch3oh -> 1 ch3 1 oh
1 ch3 1 oh -> 1 ch3oh
1 c2h6 1 h -> 1 c2h5 1 h2
1 c2h5 1 h2 -> 1 c2h6 1 h
1 ch3oh 1 ho2 -> 1 ch2oh 1 h2o2
1 ch2oh 1 h2o2 -> 1 ch3oh 1 ho2
1 c2h5 1 o2 -> 1 c2h4 1 ho2
1 c2h4 1 ho2 -> 1 c2h5 1 o2
1 c2h6 1 oh -> 1 c2h5 1 h2o
1 c2h5 1 h2o -> 1 c2h6 1 oh
1 c2h6 1 o -> 1 c2h5 1 oh
1 c2h5 1 oh -> 1 c2h6 1 o
1 ch3 1 ho2 -> 1 ch3o 1 oh
1 ch3o 1 oh -> 1 ch3 1 ho2
1 co 1 ho2 -> 1 co2 1 oh
1 co2 1 oh -> 1 co 1 ho2
1 ch3 1 ch3 -> 1 c2h6
1 c2h6 -> 1 ch3 1 ch3
1 h2o -> 1 h 1 oh
1 h 1 oh -> 1 h2o
1 h 1 o2 -> 1 ho2
1 ho2 -> 1 h 1 o2
1 co 1 o -> 1 co2
1 co2 -> 1 co 1 o
1 co 1 o2 -> 1 co2 1 o
1 co2 1 o -> 1 co 1 o2
1 hco 1 h -> 1 co 1 h2
1 co 1 h2 -> 1 hco 1 h
1 hco 1 o -> 1 co 1 oh
1 co 1 oh -> 1 hco 1 o
1 ch2o -> 1 hco 1 h
1 hco 1 h -> 1 ch2o
1 ch2o 1 oh -> 1 hco 1 h2o
1 hco 1 h2o -> 1 ch2o 1 oh
1 ch2o 1 h -> 1 hco 1 h2
1 hco 1 h2 -> 1 ch2o 1 h
1 ch2o 1 o -> 1 hco 1 oh
1 hco 1 oh -> 1 ch2o 1 o
1 ch3 1 oh -> 1 ch2o 1 h2
1 ch2o 1 h2 -> 1 ch3 1 oh
1 ch3 1 o -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch3 1 o
1 ch3 1 o2 -> 1 ch3o 1 o
1 ch3o 1 o -> 1 ch3 1 o2
1 ch2o 1 ch3 -> 1 hco 1 ch4
1 hco 1 ch4 -> 1 ch2o 1 ch3
1 hco 1 ch3 -> 1 ch4 1 co
1 ch4 1 co -> 1 hco 1 ch3
1 ch3o -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch3o
1 c2h4 -> 1 c2h2 1 h2
1 c2h2 1 h2 -> 1 c2h4
1 ho2 1 o -> 1 oh 1 o2
1 oh 1 o2 -> 1 ho2 1 o
1 hco 1 ho2 -> 1 ch2o 1 o2
1 ch2o 1 o2 -> 1 hco 1 ho2
1 ch3o 1 o2 -> 1 ch2o 1 ho2
1 ch2o 1 ho2 -> 1 ch3o 1 o2
1 ch3 1 ho2 -> 1 ch4 1 o2
1 ch4 1 o2 -> 1 ch3 1 ho2
1 hco 1 o2 -> 1 co 1 ho2
1 co 1 ho2 -> 1 hco 1 o2
1 ho2 1 h -> 1 oh 1 oh
1 oh 1 oh -> 1 ho2 1 h
1 ho2 1 h -> 1 h2 1 o2
1 h2 1 o2 -> 1 ho2 1 h
1 ho2 1 oh -> 1 h2o 1 o2
1 h2o 1 o2 -> 1 ho2 1 oh
1 h2o2 1 o2 -> 1 ho2 1 ho2
1 ho2 1 ho2 -> 1 h2o2 1 o2
1 oh 1 oh -> 1 h2o2
1 h2o2 -> 1 oh 1 oh
```

```
1 h2o2 1 h -> 1 h2o 1 oh
1 h2o 1 oh -> 1 h2o2 1 h
1 ch4 1 ho2 -> 1 ch3 1 h2o2
1 ch3 1 h2o2 -> 1 ch4 1 ho2
1 ch2o 1 ho2 -> 1 hco 1 h2o2
1 hco 1 h2o2 -> 1 ch2o 1 ho2
1 oh -> 1 o 1 h
1 o 1 h -> 1 oh
1 o2 -> 1 o 1 o
1 o 1 o -> 1 o2
1 h2 -> 1 h 1 h
1 h 1 h -> 1 h2
1 c2h3 1 h -> 1 c2h4
1 c2h4 -> 1 c2h3 1 h
1 c2h5 1 c2h3 -> 1 c2h4 1 c2h4
1 c2h4 1 c2h4 -> 1 c2h5 1 c2h3
1 c2h2 1 h -> 1 c2h3
1 c2h3 -> 1 c2h2 1 h
1 c2h4 1 h -> 1 c2h3 1 h2
1 c2h3 1 h2 -> 1 c2h4 1 h
1 c2h4 1 oh -> 1 c2h3 1 h2o
1 c2h3 1 h2o -> 1 c2h4 1 oh
1 c2h3 1 o2 -> 1 c2h2 1 ho2
1 c2h2 1 ho2 -> 1 c2h3 1 o2
1 c2h2 -> 1 c2h 1 h
1 c2h 1 h -> 1 c2h2
1 c2h2 1 o2 -> 1 hcco 1 oh
1 hcco 1 oh -> 1 c2h2 1 o2
1 ch2 1 o2 -> 1 co 1 h2o
1 co 1 h2o -> 1 ch2 1 o2
1 c2h2 1 oh -> 1 c2h 1 h2o
1 c2h 1 h2o -> 1 c2h2 1 oh
1 o 1 c2h2 -> 1 c2h 1 oh
1 c2h 1 oh -> 1 o 1 c2h2
1 c2h2 1 o -> 1 ch2 1 co
1 ch2 1 co -> 1 c2h2 1 o
-> 1 ch2
1 ch2 ->
-> 1 ch2
1 ch2 ->
-> 1 ch2
1 ch2 ->
1 c2h 1 o2 -> 1 hco 1 co
1 hco 1 co -> 1 c2h 1 o2
1 c2h 1 o -> 1 co 1 ch
1 co 1 ch -> 1 c2h 1 o
1 ch2 1 o2 -> 1 hco 1 oh
1 hco 1 oh -> 1 ch2 1 o2
1 ch2 1 o -> 1 co 1 h 1 h
1 co 1 h 1 h -> 1 ch2 1 o
1 ch2 1 h -> 1 ch 1 h2
1 ch 1 h2 -> 1 ch2 1 h
1 ch2 1 oh -> 1 ch 1 h2o
1 ch 1 h2o -> 1 ch2 1 oh
1 ch2 1 o2 -> 1 co2 1 h 1 h
1 co2 1 h 1 h -> 1 ch2 1 o2
1 ch 1 o2 -> 1 hco 1 o
1 hco 1 o -> 1 ch 1 o2
1 ch3oh 1 oh -> 1 ch2oh 1 h2o
1 ch2oh 1 h2o -> 1 ch3oh 1 oh
1 ch3oh 1 h -> 1 ch3o 1 h2
1 ch3o 1 h2 -> 1 ch3oh 1 h
1 ch3oh 1 h -> 1 ch2oh 1 h2
1 ch2oh 1 h2 -> 1 ch3oh 1 h
1 ch3oh 1 ch3 -> 1 ch2oh 1 ch4
1 ch2oh 1 ch4 -> 1 ch3oh 1 ch3
1 ch3oh 1 o -> 1 ch2oh 1 oh
1 ch2oh 1 oh -> 1 ch3oh 1 o
1 ch2oh 1 o2 -> 1 ch2o 1 ho2
1 ch2o 1 ho2 -> 1 ch2oh 1 o2
1 ch2oh -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch2oh
1 c2h3 1 o2 -> 1 c2h2 1 ho2
1 c2h2 1 ho2 -> 1 c2h3 1 o2
1 h2o2 1 o -> 1 oh 1 ho2
1 oh 1 ho2 -> 1 h2o2 1 o
1 c2h2 1 o -> 1 hcco 1 h
1 hcco 1 h -> 1 c2h2 1 o
1 c2h2 1 oh -> 1 ch2co 1 h
1 ch2co 1 h -> 1 c2h2 1 oh
1 ch2co 1 h -> 1 ch3 1 co
1 ch3 1 co -> 1 ch2co 1 h
1 ch2co 1 o -> 1 ch2 1 co2
1 ch2 1 co2 -> 1 ch2co 1 o
1 ch2 1 o2 -> 1 ch2o 1 o
1 ch2o 1 o -> 1 ch2 1 o2
1 ch2co -> 1 ch2 1 co
1 ch2 1 co -> 1 ch2co
1 ch2co 1 o -> 1 hcco 1 oh
1 hcco 1 oh -> 1 ch2co 1 o
1 ch2co 1 oh -> 1 hcco 1 h2o
1 hcco 1 h2o -> 1 ch2co 1 oh
1 ch2co 1 h -> 1 hcco 1 h2
1 hcco 1 h2 -> 1 ch2co 1 h
1 hcco 1 oh -> 1 hco 1 hco
1 hco 1 hco -> 1 hcco 1 oh
1 hcco 1 h -> 1 ch2(s) 1 co
1 ch2(s) 1 co -> 1 hcco 1 h
1 hcco 1 o -> 1 h 1 co 1 co
1 h 1 co 1 co -> 1 hcco 1 o
1 c2h6 1 o2 -> 1 c2h5 1 ho2
1 c2h5 1 ho2 -> 1 c2h6 1 o2
1 c2h6 1 ho2 -> 1 c2h5 1 h2o2
1 c2h5 1 h2o2 -> 1 c2h6 1 ho2
1 ch2 1 o2 -> 1 co2 1 h2
1 co2 1 h2 -> 1 ch2 1 o2
1 ch3 1 c2h3 -> 1 ch4 1 c2h2
1 ch4 1 c2h2 -> 1 ch3 1 c2h3
1 ch3 1 c2h5 -> 1 ch4 1 c2h4
1 ch4 1 c2h4 -> 1 ch3 1 c2h5
1 ch3oh 1 ch2o -> 1 ch3o 1 ch3o
1 ch3o 1 ch3o -> 1 ch3oh 1 ch2o
1 ch2o 1 ch3o -> 1 ch3oh 1 hco
1 ch3oh 1 hco -> 1 ch2o 1 ch3o
1 ch4 1 ch3o -> 1 ch3 1 ch3oh
1 ch3 1 ch3oh -> 1 ch4 1 ch3o
1 c2h6 1 ch3o -> 1 c2h5 1 ch3oh
1 c2h5 1 ch3oh -> 1 c2h6 1 ch3o
1 c2h3 1 h -> 1 c2h2 1 h2
1 c2h2 1 h2 -> 1 c2h3 1 h
1 ch3o 1 ch3oh -> 1 ch2oh 1 ch3oh
1 ch2oh 1 ch3oh -> 1 ch3o 1 ch3oh
1 ch3oh 1 oh -> 1 ch3o 1 h2o
1 ch3o 1 h2o -> 1 ch3oh 1 oh
1 c2h5 1 h -> 1 ch3 1 ch3
1 ch3 1 ch3 -> 1 c2h5 1 h
1 c2h3 1 o2 -> 1 ch2o 1 hco
1 ch2o 1 hco -> 1 c2h3 1 o2
1 c2h6 -> 1 c2h5 1 h
1 c2h5 1 h -> 1 c2h6
1 c2h5oh -> 1 ch2oh 1 ch3
1 ch2oh 1 ch3 -> 1 c2h5oh
1 c2h5oh -> 1 c2h5 1 oh
1 c2h5 1 oh -> 1 c2h5oh
1 c2h5oh -> 1 c2h4 1 h2o
1 c2h4 1 h2o -> 1 c2h5oh
1 c2h5oh -> 1 ch3cho 1 h2
1 ch3cho 1 h2 -> 1 c2h5oh
1 c2h5oh 1 o2 -> 1 pc2h4oh 1 ho2
1 pc2h4oh 1 ho2 -> 1 c2h5oh 1 o2
1 c2h5oh 1 o2 -> 1 sc2h4oh 1 ho2
1 sc2h4oh 1 ho2 -> 1 c2h5oh 1 o2
1 c2h5oh 1 oh -> 1 pc2h4oh 1 h2o
1 pc2h4oh 1 h2o -> 1 c2h5oh 1 oh
1 c2h5oh 1 oh -> 1 sc2h4oh 1 h2o
1 sc2h4oh 1 h2o -> 1 c2h5oh 1 oh
```

```
1 c2h5oh 1 h -> 1 pc2h4oh 1 h2
1 pc2h4oh 1 h2 -> 1 c2h5oh 1 h
1 c2h5oh 1 h -> 1 sc2h4oh 1 h2
1 sc2h4oh 1 h2 -> 1 c2h5oh 1 h
1 c2h5oh 1 ho2 -> 1 pc2h4oh 1 h2o2
1 pc2h4oh 1 h2o2 -> 1 c2h5oh 1 ho2
1 c2h5oh 1 ho2 -> 1 sc2h4oh 1 h2o2
1 sc2h4oh 1 h2o2 -> 1 c2h5oh 1 ho2
1 c2h5oh 1 ho2 -> 1 c2h5o 1 h2o2
1 c2h5o 1 h2o2 -> 1 c2h5oh 1 ho2
1 c2h5oh 1 o -> 1 pc2h4oh 1 oh
1 pc2h4oh 1 oh -> 1 c2h5oh 1 o
1 c2h5oh 1 o -> 1 sc2h4oh 1 oh
1 sc2h4oh 1 oh -> 1 c2h5oh 1 o
1 c2h5oh 1 ch3 -> 1 pc2h4oh 1 ch4
1 pc2h4oh 1 ch4 -> 1 c2h5oh 1 ch3
1 c2h5oh 1 ch3 -> 1 sc2h4oh 1 ch4
1 sc2h4oh 1 ch4 -> 1 c2h5oh 1 ch3
1 c2h5oh 1 c2h5 -> 1 pc2h4oh 1 c2h6
1 pc2h4oh 1 c2h6 -> 1 c2h5oh 1 c2h5
1 c2h5oh 1 c2h5 -> 1 sc2h4oh 1 c2h6
1 sc2h4oh 1 c2h6 -> 1 c2h5oh 1 c2h5
1 pc2h4oh -> 1 c2h4 1 oh
1 c2h4 1 oh -> 1 pc2h4oh
1 sc2h4oh -> 1 ch3cho 1 h
1 ch3cho 1 h -> 1 sc2h4oh
1 c2h4 1 ch3 -> 1 c2h3 1 ch4
1 c2h3 1 ch4 -> 1 c2h4 1 ch3
1 ch3co -> 1 ch3 1 co
1 ch3 1 co -> 1 ch3co
1 ch3cho -> 1 ch3 1 hco
1 ch3 1 hco -> 1 ch3cho
1 ch3cho 1 o2 -> 1 ch3co 1 ho2
1 ch3co 1 ho2 -> 1 ch3cho 1 o2
1 ch3cho 1 oh -> 1 ch3co 1 h2o
1 ch3co 1 h2o -> 1 ch3cho 1 oh
1 ch3cho 1 h -> 1 ch3co 1 h2
1 ch3co 1 h2 -> 1 ch3cho 1 h
1 ch3cho 1 o -> 1 ch3co 1 oh
1 ch3co 1 oh -> 1 ch3cho 1 o
1 ch3cho 1 ho2 -> 1 ch3co 1 h2o2
1 ch3co 1 h2o2 -> 1 ch3cho 1 ho2
1 ch3cho 1 ch3 -> 1 ch3co 1 ch4
1 ch3co 1 ch4 -> 1 ch3cho 1 ch3
1 c2h4 1 o2 -> 1 c2h3 1 ho2
1 c2h3 1 ho2 -> 1 c2h4 1 o2
1 ch2o -> 1 co 1 h2
1 co 1 h2 -> 1 ch2o
1 c2h4 1 ch3o -> 1 c2h3 1 ch3oh
1 c2h3 1 ch3oh -> 1 c2h4 1 ch3o
1 ch3coch3 -> 1 ch3co 1 ch3
1 ch3co 1 ch3 -> 1 ch3coch3
1 ch3coch3 1 oh -> 1 ch3coch2 1 h2o
1 ch3coch2 1 h2o -> 1 ch3coch3 1 oh
1 ch3coch3 1 h -> 1 ch3coch2 1 h2
1 ch3coch2 1 h2 -> 1 ch3coch3 1 h
1 ch3coch3 1 o -> 1 ch3coch2 1 oh
1 ch3coch2 1 oh -> 1 ch3coch3 1 o
1 ch3coch3 1 ch3 -> 1 ch3coch2 1 ch4
1 ch3coch2 1 ch4 -> 1 ch3coch3 1 ch3
1 ch3coch3 1 ch3o -> 1 ch3coch2 1 ch3oh
1 ch3coch2 1 ch3oh -> 1 ch3coch3 1 ch3o
1 ch3coch2 -> 1 ch2co 1 ch3
1 ch2co 1 ch3 -> 1 ch3coch2
1 ch3coch3 1 o2 -> 1 ch3coch2 1 ho2
1 ch3coch2 1 ho2 -> 1 ch3coch3 1 o2
1 ch3coch3 1 ho2 -> 1 ch3coch2 1 h2o2
1 ch3coch2 1 h2o2 -> 1 ch3coch3 1 ho2
1 c2h5co -> 1 c2h5 1 co
1 c2h5 1 co -> 1 c2h5co
1 c2h5cho 1 h -> 1 c2h5co 1 h2
1 c2h5co 1 h2 -> 1 c2h5cho 1 h
1 c2h5cho 1 o -> 1 c2h5co 1 oh
1 c2h5co 1 oh -> 1 c2h5cho 1 o
1 c2h5cho 1 oh -> 1 c2h5co 1 h2o
1 c2h5co 1 h2o -> 1 c2h5cho 1 oh
1 c2h5cho 1 ch3 -> 1 c2h5co 1 ch4
1 c2h5co 1 ch4 -> 1 c2h5cho 1 ch3
1 c2h5cho 1 ho2 -> 1 c2h5co 1 h2o2
1 c2h5co 1 h2o2 -> 1 c2h5cho 1 ho2
1 c2h5cho 1 ch3o -> 1 c2h5co 1 ch3oh
1 c2h5co 1 ch3oh -> 1 c2h5cho 1 ch3o
1 c2h5cho 1 c2h5 -> 1 c2h5co 1 c2h6
1 c2h5co 1 c2h6 -> 1 c2h5cho 1 c2h5
1 c2h5cho -> 1 c2h5 1 hco
1 c2h5 1 hco -> 1 c2h5cho
1 c2h5cho 1 o2 -> 1 c2h5co 1 ho2
1 c2h5co 1 ho2 -> 1 c2h5cho 1 o2
1 c2h5cho 1 c2h3 -> 1 c2h5co 1 c2h4
1 c2h5co 1 c2h4 -> 1 c2h5cho 1 c2h3
1 h2o2 1 h -> 1 h2 1 ho2
1 h2 1 ho2 -> 1 h2o2 1 h
1 hco 1 o -> 1 co2 1 h
1 co2 1 h -> 1 hco 1 o
1 ch3 -> 1 ch2 1 h
1 ch2 1 h -> 1 ch3
1 ch3 1 h -> 1 ch2 1 h2
1 ch2 1 h2 -> 1 ch3 1 h
1 ch3 1 oh -> 1 ch2 1 h2o
1 ch2 1 h2o -> 1 ch3 1 oh
1 ch 1 ch4 -> 1 c2h4 1 h
1 c2h4 1 h -> 1 ch 1 ch4
1 ch3oh -> 1 ch2oh 1 h
1 ch2oh 1 h -> 1 ch3oh
1 ch3co 1 h -> 1 ch2co 1 h2
1 ch2co 1 h2 -> 1 ch3co 1 h
1 ch3co 1 o -> 1 ch2co 1 oh
1 ch2co 1 oh -> 1 ch3co 1 o
1 ch3co 1 ch3 -> 1 ch2co 1 ch4
1 ch2co 1 ch4 -> 1 ch3co 1 ch3
1 c2h4 1 o -> 1 ch2cho 1 h
1 ch2cho 1 h -> 1 c2h4 1 o
1 c2h5 1 o -> 1 ch3cho 1 h
1 ch3cho 1 h -> 1 c2h5 1 o
1 c2h6 1 ch -> 1 c2h5 1 ch2
1 c2h5 1 ch2 -> 1 c2h6 1 ch
1 c2h5oh 1 oh -> 1 c2h5o 1 h2o
1 c2h5o 1 h2o -> 1 c2h5oh 1 oh
1 c2h5oh 1 h -> 1 c2h5o 1 h2
1 c2h5o 1 h2 -> 1 c2h5oh 1 h
1 c2h5oh 1 o -> 1 c2h5o 1 oh
1 c2h5o 1 oh -> 1 c2h5oh 1 o
1 c2h5oh 1 ch3 -> 1 c2h5o 1 ch4
1 c2h5o 1 ch4 -> 1 c2h5oh 1 ch3
1 sc2h4oh 1 o2 -> 1 ch3cho 1 ho2
1 ch3cho 1 ho2 -> 1 sc2h4oh 1 o2
1 c2h5o 1 o2 -> 1 ch3cho 1 ho2
1 ch3cho 1 ho2 -> 1 c2h5o 1 o2
1 h2o2 1 o2 -> 1 ho2 1 ho2
1 ho2 1 ho2 -> 1 h2o2 1 o2
1 h2o2 1 oh -> 1 h2o 1 ho2
1 h2o 1 ho2 -> 1 h2o2 1 oh
1 c2h5o2 -> 1 c2h5 1 o2
1 c2h5 1 o2 -> 1 c2h5o2
1 ch3o2 -> 1 ch3 1 o2
1 ch3 1 o2 -> 1 ch3o2
1 ch3o2h -> 1 ch3o 1 oh
1 ch3o 1 oh -> 1 ch3o2h
1 c2h5o2h -> 1 c2h5o 1 oh
1 c2h5o 1 oh -> 1 c2h5o2h
1 c2h5o -> 1 ch3 1 ch2o
1 ch3 1 ch2o -> 1 c2h5o
1 ch3o2 1 ch2o -> 1 ch3o2h 1 hco
1 ch3o2h 1 hco -> 1 ch3o2 1 ch2o
1 c2h5o2 1 ch2o -> 1 c2h5o2h 1 hco
1 c2h5o2h 1 hco -> 1 c2h5o2 1 ch2o
```

```
1 c2h4 1 ch3o2 -> 1 c2h3 1 ch3o2h
1 c2h3 1 ch3o2h -> 1 c2h4 1 ch3o2
1 c2h4 1 c2h5o2 -> 1 c2h3 1 c2h5o2h
1 c2h3 1 c2h5o2h -> 1 c2h4 1 c2h5o2
1 ch4 1 ch3o2 -> 1 ch3 1 ch3o2h
1 ch3 1 ch3o2h -> 1 ch4 1 ch3o2
1 ch4 1 c2h5o2 -> 1 ch3 1 c2h5o2h
1 ch3 1 c2h5o2h -> 1 ch4 1 c2h5o2
1 ch3oh 1 ch3o2 -> 1 ch2oh 1 ch3o2h
1 ch2oh 1 ch3o2h -> 1 ch3oh 1 ch3o2
1 ch3oh 1 c2h5o2 -> 1 ch2oh 1 c2h5o2h
1 ch2oh 1 c2h5o2h -> 1 ch3oh 1 c2h5o2
1 c2h5 1 ho2 -> 1 c2h5o 1 oh
1 c2h5o 1 oh -> 1 c2h5 1 ho2
1 ch3o2 1 ch3 -> 1 ch3o 1 ch3o
1 ch3o 1 ch3o -> 1 ch3o2 1 ch3
1 ch3o2 1 c2h5 -> 1 ch3o 1 c2h5o
1 ch3o 1 c2h5o -> 1 ch3o2 1 c2h5
1 ch3o2 1 ho2 -> 1 ch3o2h 1 o2
1 ch3o2h 1 o2 -> 1 ch3o2 1 ho2
1 ch3oh 1 o2 -> 1 ch2oh 1 ho2
1 ch2oh 1 ho2 -> 1 ch3oh 1 o2
1 c2h5o2 1 ho2 -> 1 c2h5o2h 1 o2
1 c2h5o2h 1 o2 -> 1 c2h5o2 1 ho2
1 ch3o2 1 ch3o2 -> 1 ch2o 1 ch3oh 1 o2
1 ch2o 1 ch3oh 1 o2 -> 1 ch3o2 1 ch3o2
1 ch3o2 1 ch3o2 -> 1 o2 1 ch3o 1 ch3o
1 o2 1 ch3o 1 ch3o -> 1 ch3o2 1 ch3o2
1 c2h6 1 ch3o2 -> 1 c2h5 1 ch3o2h
1 c2h5 1 ch3o2h -> 1 c2h6 1 ch3o2
1 c2h6 1 c2h5o2 -> 1 c2h5 1 c2h5o2h
1 c2h5 1 c2h5o2h -> 1 c2h6 1 c2h5o2
1 o2c2h4oh -> 1 pc2h4oh 1 o2
1 pc2h4oh 1 o2 -> 1 o2c2h4oh
1 o2c2h4oh -> 1 oh 1 ch2o 1 ch2o
1 oh 1 ch2o 1 ch2o -> 1 o2c2h4oh
1 c2h5o2 -> 1 c2h4o2h
1 c2h4o2h -> 1 c2h5o2
1 c2h4o2h -> 1 c2h4o1-2 1 oh
1 c2h4o1-2 1 oh -> 1 c2h4o2h
1 ch3co3 -> 1 ch3co 1 o2
1 ch3co 1 o2 -> 1 ch3co3
1 ch3co2 -> 1 ch3 1 co2
1 ch3 1 co2 -> 1 ch3co2
1 ch3co3h -> 1 ch3co2 1 oh
1 ch3co2 1 oh -> 1 ch3co3h
1 ch3co3 1 ho2 -> 1 ch3co3h 1 o2
1 ch3co3h 1 o2 -> 1 ch3co3 1 ho2
1 c2h5o -> 1 ch3cho 1 h
1 ch3cho 1 h -> 1 c2h5o
1 h2o2 1 ch3co3 -> 1 ho2 1 ch3co3h
1 ho2 1 ch3co3h -> 1 h2o2 1 ch3co3
1 ch4 1 ch3co3 -> 1 ch3 1 ch3co3h
1 ch3 1 ch3co3h -> 1 ch4 1 ch3co3
1 c2h4 1 ch3co3 -> 1 c2h3 1 ch3co3h
1 c2h3 1 ch3co3h -> 1 c2h4 1 ch3co3
1 c2h6 1 ch3co3 -> 1 c2h5 1 ch3co3h
1 c2h5 1 ch3co3h -> 1 c2h6 1 ch3co3
1 ch2o 1 ch3co3 -> 1 hco 1 ch3co3h
1 hco 1 ch3co3h -> 1 ch2o 1 ch3co3
1 ch3o2 1 ch3cho -> 1 ch3o2h 1 ch3co
1 ch3o2h 1 ch3co -> 1 ch3o2 1 ch3cho
1 ch3cho 1 ch3co3 -> 1 ch3co 1 ch3co3h
1 ch3co 1 ch3co3h -> 1 ch3cho 1 ch3co3
1 c2h3co -> 1 c2h3 1 co
1 c2h3 1 co -> 1 c2h3co
1 c2h3cho 1 oh -> 1 c2h3co 1 h2o
1 c2h3co 1 h2o -> 1 c2h3cho 1 oh
1 c2h3cho 1 h -> 1 c2h3co 1 h2
1 c2h3co 1 h2 -> 1 c2h3cho 1 h
1 c2h3cho 1 o -> 1 c2h3co 1 oh
1 c2h3co 1 oh -> 1 c2h3cho 1 o
1 c2h3cho 1 ho2 -> 1 c2h3co 1 h2o2
1 c2h3co 1 h2o2 -> 1 c2h3cho 1 ho2
1 c2h3cho 1 ch3 -> 1 c2h3co 1 ch4
1 c2h3co 1 ch4 -> 1 c2h3cho 1 ch3
1 c2h3cho 1 ch3o2 -> 1 c2h3co 1 ch3o2h
1 c2h3co 1 ch3o2h -> 1 c2h3cho 1 ch3o2
1 c2h4o2h -> 1 c2h4 1 ho2
1 c2h4 1 ho2 -> 1 c2h4o2h
1 c2h4 1 ch3o2 -> 1 c2h4o1-2 1 ch3o
1 c2h4o1-2 1 ch3o -> 1 c2h4 1 ch3o2
1 c2h4 1 c2h5o2 -> 1 c2h4o1-2 1 c2h5o
1 c2h4o1-2 1 c2h5o -> 1 c2h4 1 c2h5o2
1 c2h4o1-2 -> 1 ch3 1 hco
1 ch3 1 hco -> 1 c2h4o1-2
-> 1 ch3
1 ch3 ->
1 c2h4o1-2 -> 1 ch3cho
1 ch3cho -> 1 c2h4o1-2
1 c2h4o1-2 1 oh -> 1 c2h3o1-2 1 h2o
1 c2h3o1-2 1 h2o -> 1 c2h4o1-2 1 oh
1 c2h4o1-2 1 h -> 1 c2h3o1-2 1 h2
1 c2h3o1-2 1 h2 -> 1 c2h4o1-2 1 h
1 c2h4o1-2 1 ho2 -> 1 c2h3o1-2 1 h2o2
1 c2h3o1-2 1 h2o2 -> 1 c2h4o1-2 1 ho2
1 c2h4o1-2 1 ch3o2 -> 1 c2h3o1-2 1 ch3o2h
1 c2h3o1-2 1 ch3o2h -> 1 c2h4o1-2 1 ch3o2
1 c2h4o1-2 1 c2h5o2 -> 1 c2h3o1-2 1 c2h5o2h
1 c2h3o1-2 1 c2h5o2h -> 1 c2h4o1-2 1 c2h5o2
1 c2h4o1-2 1 ch3 -> 1 c2h3o1-2 1 ch4
1 c2h3o1-2 1 ch4 -> 1 c2h4o1-2 1 ch3
1 c2h4o1-2 1 ch3o -> 1 c2h3o1-2 1 ch3oh
1 c2h3o1-2 1 ch3oh -> 1 c2h4o1-2 1 ch3o
1 ch3coch2o2 -> 1 ch3coch2 1 o2
1 ch3coch2 1 o2 -> 1 ch3coch2o2
1 ch3coch3 1 ch3coch2o2 -> 1 ch3coch2 1 ch3coch2o2h
1 ch3coch2 1 ch3coch2o2h -> 1 ch3coch3 1 ch3coch2o2
1 ch2o 1 ch3coch2o2 -> 1 hco 1 ch3coch2o2h
1 hco 1 ch3coch2o2h -> 1 ch2o 1 ch3coch2o2
1 ho2 1 ch3coch2o2 -> 1 ch3coch2o2h 1 o2
1 ch3coch2o2h 1 o2 -> 1 ho2 1 ch3coch2o2
1 ch3coch2o2h -> 1 ch3coch2o 1 oh
1 ch3coch2o 1 oh -> 1 ch3coch2o2h
1 ch3coch2o -> 1 ch3co 1 ch2o
1 ch3co 1 ch2o -> 1 ch3coch2o
1 c2h5cho 1 ch3o2 -> 1 c2h5co 1 ch3o2h
1 c2h5co 1 ch3o2h -> 1 c2h5cho 1 ch3o2
1 c2h5cho 1 c2h5o -> 1 c2h5co 1 c2h5oh
1 c2h5co 1 c2h5oh -> 1 c2h5cho 1 c2h5o
1 c2h5cho 1 c2h5o2 -> 1 c2h5co 1 c2h5o2h
1 c2h5co 1 c2h5o2h -> 1 c2h5cho 1 c2h5o2
1 c2h5cho 1 ch3co3 -> 1 c2h5co 1 ch3co3h
1 c2h5co 1 ch3co3h -> 1 c2h5cho 1 ch3co3
1 ch3cho 1 oh -> 1 ch3
1 ch3 -> 1 ch3cho 1 oh
1 c2h3o1-2 -> 1 ch3co
1 ch3co -> 1 c2h3o1-2
1 c2h3o1-2 -> 1 ch2cho
1 ch2cho -> 1 c2h3o1-2
1 ch2cho -> 1 ch2co 1 h
1 ch2co 1 h -> 1 ch2cho
1 ch2cho 1 o2 -> 1 ch2o 1 co 1 oh
1 ch2o 1 co 1 oh -> 1 ch2cho 1 o2
1 hco3 -> 1 hco 1 o2
1 hco 1 o2 -> 1 hco3
1 ch2o 1 hco3 -> 1 hco 1 hco3h
1 hco 1 hco3h -> 1 ch2o 1 hco3
1 hco3h -> 1 hco2 1 oh
1 hco2 1 oh -> 1 hco3h
1 hco2 -> 1 h 1 co2
1 h 1 co2 -> 1 hco2
1 hcco 1 o2 -> 1 co2 1 hco
1 co2 1 hco -> 1 hcco 1 o2
1 ch3cho 1 oh -> 1 ch2cho
1 ch2cho -> 1 ch3cho 1 oh
```

```
1 ch2co 1 oh -> 1 ch2oh 1 co
1 ch2oh 1 co -> 1 ch2co 1 oh
1 ch3 1 o2 -> 1 ch2o 1 oh
1 ch2o 1 oh -> 1 ch3 1 o2
1 c2h4 1 h2 -> 1 ch3 1 ch3
1 ch3 1 ch3 -> 1 c2h4 1 h2
1 ch3 1 oh -> 1 ch2(s) 1 h2o
1 ch2(s) 1 h2o -> 1 ch3 1 oh
1 c2h4 1 ho2 -> 1 c2h4o1-2 1 oh
1 c2h4o1-2 1 oh -> 1 c2h4 1 ho2
1 ch3och3 -> 1 ch3 1 ch3o
1 ch3 1 ch3o -> 1 ch3och3
1 ch3och3 1 oh -> 1 ch3och2 1 h2o
1 ch3och2 1 h2o -> 1 ch3och3 1 oh
1 ch3och3 1 h -> 1 ch3och2 1 h2
1 ch3och2 1 h2 -> 1 ch3och3 1 h
1 ch3och3 1 o -> 1 ch3och2 1 oh
1 ch3och2 1 oh -> 1 ch3och3 1 o
1 ch3och3 1 ho2 -> 1 ch3och2 1 h2o2
1 ch3och2 1 h2o2 -> 1 ch3och3 1 ho2
1 ch3och3 1 ch3o2 -> 1 ch3och2 1 ch3o2h
1 ch3och2 1 ch3o2h -> 1 ch3och3 1 ch3o2
1 ch3och3 1 ch3 -> 1 ch3och2 1 ch4
1 ch3och2 1 ch4 -> 1 ch3och3 1 ch3
1 ch3och3 1 o2 -> 1 ch3och2 1 ho2
1 ch3och2 1 ho2 -> 1 ch3och3 1 o2
1 ch3och3 1 ch3o -> 1 ch3och2 1 ch3oh
1 ch3och2 1 ch3oh -> 1 ch3och3 1 ch3o
1 ch3och2 -> 1 ch2o 1 ch3
1 ch2o 1 ch3 -> 1 ch3och2
1 ch3och2 1 ch3o -> 1 ch3och3 1 ch2o
1 ch3och3 1 ch2o -> 1 ch3och2 1 ch3o
1 ch3och2 1 ch2o -> 1 ch3och3 1 hco
1 ch3och3 1 hco -> 1 ch3och2 1 ch2o
1 ch3och2 1 ch3cho -> 1 ch3och3 1 ch3co
1 ch3och3 1 ch3co -> 1 ch3och2 1 ch3cho
1 ch3och2 1 ho2 -> 1 ch3och2o 1 oh
1 ch3och2o 1 oh -> 1 ch3och2 1 ho2
1 ch3och2o2 -> 1 ch3och2 1 o2
1 ch3och2 1 o2 -> 1 ch3och2o2
1 ch3och3 1 ch3och2o2 -> 1 ch3och2 1 ch3och2o2h
1 ch3och2 1 ch3och2o2h -> 1 ch3och3 1 ch3och2o2
1 ch3och2o2 1 ch2o -> 1 ch3och2o2h 1 hco
1 ch3och2o2h 1 hco -> 1 ch3och2o2 1 ch2o
1 ch3och2o2 1 ch3cho -> 1 ch3och2o2h 1 ch3co
1 ch3och2o2h 1 ch3co -> 1 ch3och2o2 1 ch3cho
1 ch3och2o2h -> 1 ch3och2o 1 oh
1 ch3och2o 1 oh -> 1 ch3och2o2h
1 ch3och2o -> 1 ch3o 1 ch2o
1 ch3o 1 ch2o -> 1 ch3och2o
1 ch3och2o2 -> 1 ch2och2o2h
1 ch2och2o2h -> 1 ch3och2o2
1 ch2och2o2h -> 1 oh 1 ch2o 1 ch2o
1 oh 1 ch2o 1 ch2o -> 1 ch2och2o2h
1 o2ch2och2o2h -> 1 ch2och2o2h 1 o2
1 ch2och2o2h 1 o2 -> 1 o2ch2och2o2h
1 o2ch2och2o2h -> 1 ho2ch2ocho 1 oh
1 ho2ch2ocho 1 oh -> 1 o2ch2och2o2h
1 ho2ch2ocho -> 1 och2ocho 1 oh
1 och2ocho 1 oh -> 1 ho2ch2ocho
1 och2ocho -> 1 ch2o 1 hco2
1 ch2o 1 hco2 -> 1 och2ocho
1 c2h5o2 -> 1 c2h4 1 ho2
1 c2h4 1 ho2 -> 1 c2h5o2
1 c2h4o2h -> 1 c2h5 1 o2
1 c2h5 1 o2 -> 1 c2h4o2h
1 ch3o 1 ch3 -> 1 ch2o 1 ch4
1 ch2o 1 ch4 -> 1 ch3o 1 ch3
1 ch3och3 1 hco3 -> 1 ch3och2 1 hco3h
1 ch3och2 1 hco3h -> 1 ch3och3 1 hco3
1 och2ocho -> 1 hoch2oco
1 hoch2oco -> 1 och2ocho
1 hoch2oco -> 1 hoch2o 1 co
1 hoch2o 1 co -> 1 hoch2oco
1 hoch2oco -> 1 ch2oh 1 co2
1 ch2oh 1 co2 -> 1 hoch2oco
1 ch2oh 1 ho2 -> 1 hoch2o 1 oh
1 hoch2o 1 oh -> 1 ch2oh 1 ho2
1 hoch2o -> 1 ch2o 1 oh
1 ch2o 1 oh -> 1 hoch2o
1 hoch2o -> 1 hco2h 1 h
1 hco2h 1 h -> 1 hoch2o
1 hco2h -> 1 co 1 h2o
1 co 1 h2o -> 1 hco2h
1 hco2h -> 1 co2 1 h2
1 co2 1 h2 -> 1 hco2h
1 ch3och2o2 1 ch3och2o2 -> 1 ch3ocho 1 ch3och2oh 1 o2
1 ch3ocho 1 ch3och2oh 1 o2 -> 1 ch3och2o2 1 ch3och2o2
1 ch3och2o2 1 ch3och2o2 -> 1 o2 1 ch3och2o 1 ch3och2o
1 o2 1 ch3och2o 1 ch3och2o -> 1 ch3och2o2 1 ch3och2o2
1 ch3och2o -> 1 ch3ocho 1 h
1 ch3ocho 1 h -> 1 ch3och2o
1 ch3ocho -> 1 ch3 1 hco2
1 ch3 1 hco2 -> 1 ch3ocho
1 ch3ocho 1 o2 -> 1 ch3oco 1 ho2
1 ch3oco 1 ho2 -> 1 ch3ocho 1 o2
1 ch3ocho 1 oh -> 1 ch3oco 1 h2o
1 ch3oco 1 h2o -> 1 ch3ocho 1 oh
1 ch3ocho 1 ho2 -> 1 ch3oco 1 h2o2
1 ch3oco 1 h2o2 -> 1 ch3ocho 1 ho2
1 ch3ocho 1 o -> 1 ch3oco 1 oh
1 ch3oco 1 oh -> 1 ch3ocho 1 o
1 ch3ocho 1 h -> 1 ch3oco 1 h2
1 ch3oco 1 h2 -> 1 ch3ocho 1 h
1 ch3ocho 1 ch3 -> 1 ch3oco 1 ch4
1 ch3oco 1 ch4 -> 1 ch3ocho 1 ch3
1 ch3ocho 1 ch3o -> 1 ch3oco 1 ch3oh
1 ch3oco 1 ch3oh -> 1 ch3ocho 1 ch3o
1 ch3ocho 1 ch3o2 -> 1 ch3oco 1 ch3o2h
1 ch3oco 1 ch3o2h -> 1 ch3ocho 1 ch3o2
1 ch3oco -> 1 ch3o 1 co
1 ch3o 1 co -> 1 ch3oco
1 ch3oco -> 1 ch3 1 co2
1 ch3 1 co2 -> 1 ch3oco
1 och2o2h -> 1 ch2o 1 ho2
1 ch2o 1 ho2 -> 1 och2o2h
1 och2o2h -> 1 hoch2o2
1 hoch2o2 -> 1 och2o2h
1 hoch2o2 1 ho2 -> 1 hoch2o2h 1 o2
1 hoch2o2h 1 o2 -> 1 hoch2o2 1 ho2
1 ch3och3 1 hco2 -> 1 ch3och2 1 hco2h
1 ch3och2 1 hco2h -> 1 ch3och3 1 hco2
1 hco2h -> 1 hco 1 oh
1 hco 1 oh -> 1 hco2h
1 ch2o 1 hco2 -> 1 hco 1 hco2h
1 hco 1 hco2h -> 1 ch2o 1 hco2
1 hco2 1 ho2 -> 1 hco2h 1 o2
1 hco2h 1 o2 -> 1 hco2 1 ho2
1 hco2 1 h2o2 -> 1 hco2h 1 ho2
1 hco2h 1 ho2 -> 1 hco2 1 h2o2
1 hco2h 1 oh -> 1 h2o 1 co2 1 h
1 h2o 1 co2 1 h -> 1 hco2h 1 oh
1 hco2h 1 oh -> 1 h2o 1 co 1 oh
1 h2o 1 co 1 oh -> 1 hco2h 1 oh
1 hco2h 1 h -> 1 h2 1 co2 1 h
1 h2 1 co2 1 h -> 1 hco2h 1 h
1 hco2h 1 h -> 1 h2 1 co 1 oh
1 h2 1 co 1 oh -> 1 hco2h 1 h
1 hco2h 1 ch3 -> 1 ch4 1 co 1 oh
1 ch4 1 co 1 oh -> 1 hco2h 1 ch3
1 hco2h 1 ho2 -> 1 h2o2 1 co 1 oh
1 h2o2 1 co 1 oh -> 1 hco2h 1 ho2
1 hco2h 1 o -> 1 co 1 oh 1 oh
1 co 1 oh 1 oh -> 1 hco2h 1 o
1 ch2(s) -> 1 ch2
1 ch2 -> 1 ch2(s)
```

```
1 ch2(s) 1 ch4 -> 1 ch3 1 ch3
1 ch3 1 ch3 -> 1 ch2(s) 1 ch4
1 ch2(s) 1 c2h6 -> 1 ch3 1 c2h5
1 ch3 1 c2h5 -> 1 ch2(s) 1 c2h6
1 ch2(s) 1 o2 -> 1 co 1 oh 1 h
1 co 1 oh 1 h -> 1 ch2(s) 1 o2
1 ch2(s) 1 h2 -> 1 ch3 1 h
1 ch3 1 h -> 1 ch2(s) 1 h2
1 ch2(s) 1 h -> 1 ch 1 h2
1 ch 1 h2 -> 1 ch2(s) 1 h
1 ch2(s) 1 o -> 1 co 1 h 1 h
1 co 1 h 1 h -> 1 ch2(s) 1 o
1 ch2(s) 1 oh -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch2(s) 1 oh
1 ch2(s) 1 co2 -> 1 ch2o 1 co
1 ch2o 1 co -> 1 ch2(s) 1 co2
1 ch2(s) 1 ch3 -> 1 c2h4 1 h
1 c2h4 1 h -> 1 ch2(s) 1 ch3
1 ch2(s) 1 ch2co -> 1 c2h4 1 co
1 c2h4 1 co -> 1 ch2(s) 1 ch2co
1 c2h3 1 o2 -> 1 ch2cho 1 o
1 ch2cho 1 o -> 1 c2h3 1 o2
```

E 2.2 Ethanol

```
# Number of Components:
57
# Components:
h2
h
ch4
ch3
ch2
ch
ch2o
hco
co2
co
o2
o
oh
ho2
h2o2
h2o
c2h
hcco
c2h2
c2h3
c2h4
c2h5
c2h6
ch2oh
ch3o
hccoh
h2ccch
c3h2
ch2(s)
ch2co
c2o
hcoh
ch3oh
ch2hco
c3h6
ac3h5
pc3h5
sc3h5
ch2chcho
pc3h4
ac3h4
ch3co
ch2chco
ch3chco
ch3hco
chocho
ic3h7
nc3h7
c2h5oh
c2h4oh
ch3choh
ch3ch2o
ch2chch2o
hcooh
c3h8
hoc2h4o2
n2
# Number of Reactions:
752
# Reactions:
1 oh 1 h2 -> 1 h 1 h2o
1 h 1 h2o -> 1 oh 1 h2
1 o 1 oh -> 1 o2 1 h
1 o2 1 h -> 1 o 1 oh
1 o 1 h2 -> 1 oh 1 h
1 oh 1 h -> 1 o 1 h2
1 h 1 o2 -> 1 ho2
1 ho2 -> 1 h 1 o2
1 h 1 o2 -> 1 ho2
1 ho2 -> 1 h 1 o2
1 h 1 o2 -> 1 ho2
1 ho2 -> 1 h 1 o2
1 h 1 o2 -> 1 ho2
1 ho2 -> 1 h 1 o2
1 oh 1 ho2 -> 1 h2o 1 o2
1 h2o 1 o2 -> 1 oh 1 ho2
1 h 1 ho2 -> 1 oh 1 oh
1 oh 1 oh -> 1 h 1 ho2
1 h 1 ho2 -> 1 h2 1 o2
1 h2 1 o2 -> 1 h 1 ho2
1 h 1 ho2 -> 1 o 1 h2o
1 o 1 h2o -> 1 h 1 ho2
1 o 1 ho2 -> 1 o2 1 oh
1 o2 1 oh -> 1 o 1 ho2
1 oh 1 oh -> 1 o 1 h2o
1 o 1 h2o -> 1 oh 1 oh
1 h 1 h -> 1 h2
1 h2 -> 1 h 1 h
1 h 1 h 1 h2 -> 1 h2 1 h2
1 h2 1 h2 -> 1 h 1 h 1 h2
1 h 1 h 1 h2o -> 1 h2 1 h2o
1 h2 1 h2o -> 1 h 1 h 1 h2o
1 h 1 oh -> 1 h2o
1 h2o -> 1 h 1 oh
1 h 1 o -> 1 oh
1 oh -> 1 h 1 o
1 o 1 o -> 1 o2
1 o2 -> 1 o 1 o
1 ho2 1 ho2 -> 1 h2o2 1 o2
1 h2o2 1 o2 -> 1 ho2 1 ho2
1 oh 1 oh -> 1 h2o2
1 h2o2 -> 1 oh 1 oh
1 h2o2 1 h -> 1 ho2 1 h2
1 ho2 1 h2 -> 1 h2o2 1 h
1 h2o2 1 h -> 1 oh 1 h2o
1 oh 1 h2o -> 1 h2o2 1 h
1 h2o2 1 o -> 1 oh 1 ho2
1 oh 1 ho2 -> 1 h2o2 1 o
1 h2o2 1 oh -> 1 h2o 1 ho2
1 h2o 1 ho2 -> 1 h2o2 1 oh
1 ch3 1 ch3 -> 1 c2h6
1 c2h6 -> 1 ch3 1 ch3
1 ch3 1 h -> 1 ch4
1 ch4 -> 1 ch3 1 h
1 ch4 1 h -> 1 ch3 1 h2
1 ch3 1 h2 -> 1 ch4 1 h
1 ch4 1 oh -> 1 ch3 1 h2o
```

```
1 ch3 1 h2o -> 1 ch4 1 oh
1 ch4 1 o -> 1 ch3 1 oh
1 ch3 1 oh -> 1 ch4 1 o
1 ch4 1 ho2 -> 1 ch3 1 h2o2
1 ch3 1 h2o2 -> 1 ch4 1 ho2
1 ch3 1 ho2 -> 1 ch3o 1 oh
1 ch3o 1 oh -> 1 ch3 1 ho2
1 ch3 1 ho2 -> 1 ch4 1 o
1 ch4 1 o -> 1 ch3 1 ho2
1 ch3 1 o -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch3 1 o
1 ch3 1 o2 -> 1 ch3o 1 o
1 ch3o 1 o -> 1 ch3 1 o2
1 ch3 1 o2 -> 1 ch2o 1 oh
1 ch2o 1 oh -> 1 ch3 1 o2
1 ch3o 1 h -> 1 ch3 1 oh
1 ch3 1 oh -> 1 ch3o 1 h
1 ch2oh 1 h -> 1 ch3 1 oh
1 ch3 1 oh -> 1 ch2oh 1 h
1 ch3 1 oh -> 1 ch2(s) 1 h2o
1 ch2(s) 1 h2o -> 1 ch3 1 oh
1 ch3 1 oh -> 1 hcoh 1 h2
1 hcoh 1 h2 -> 1 ch3 1 oh
1 ch3 1 oh -> 1 ch2 1 h2o
1 ch2 1 h2o -> 1 ch3 1 oh
1 ch3 1 h -> 1 ch2 1 h2
1 ch2 1 h2 -> 1 ch3 1 h
1 ch3 -> 1 ch 1 h2
1 ch 1 h2 -> 1 ch3
1 ch3 -> 1 ch2 1 h
1 ch2 1 h -> 1 ch3
1 ch3 1 oh -> 1 ch3oh
1 ch3oh -> 1 ch3 1 oh
1 ch3oh -> 1 ch2(s) 1 h2o
1 ch2(s) 1 h2o -> 1 ch3oh
1 ch3oh -> 1 hcoh 1 h2
1 hcoh 1 h2 -> 1 ch3oh
1 ch3oh -> 1 ch2o 1 h2
1 ch2o 1 h2 -> 1 ch3oh
1 ch3oh 1 oh -> 1 ch2oh 1 h2o
1 ch2oh 1 h2o -> 1 ch3oh 1 oh
1 ch3oh 1 oh -> 1 ch3o 1 h2o
1 ch3o 1 h2o -> 1 ch3oh 1 oh
1 ch3oh 1 o -> 1 ch2oh 1 oh
1 ch2oh 1 oh -> 1 ch3oh 1 o
1 ch3oh 1 h -> 1 ch2oh 1 h2
1 ch2oh 1 h2 -> 1 ch3oh 1 h
1 ch3oh 1 h -> 1 ch3o 1 h2
1 ch3o 1 h2 -> 1 ch3oh 1 h
1 ch3oh 1 ch3 -> 1 ch2oh 1 ch4
1 ch2oh 1 ch4 -> 1 ch3oh 1 ch3
1 ch3oh 1 ch3 -> 1 ch3o 1 ch4
1 ch3o 1 ch4 -> 1 ch3oh 1 ch3
1 ch3oh 1 ho2 -> 1 ch2oh 1 h2o2
1 ch2oh 1 h2o2 -> 1 ch3oh 1 ho2
1 ch2o 1 h -> 1 ch3o
1 ch3o -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch2oh
1 ch2oh -> 1 ch2o 1 h
1 ch3o 1 ch3 -> 1 ch2o 1 ch4
1 ch2o 1 ch4 -> 1 ch3o 1 ch3
1 ch3o 1 h -> 1 ch2o 1 h2
1 ch2o 1 h2 -> 1 ch3o 1 h
1 ch2oh 1 h -> 1 ch2o 1 h2
1 ch2o 1 h2 -> 1 ch2oh 1 h
1 ch3o 1 oh -> 1 ch2o 1 h2o
1 ch2o 1 h2o -> 1 ch3o 1 oh
1 ch2oh 1 oh -> 1 ch2o 1 h2o
1 ch2o 1 h2o -> 1 ch2oh 1 oh
1 ch3o 1 o -> 1 ch2o 1 oh
1 ch2o 1 oh -> 1 ch3o 1 o
1 ch2oh 1 o -> 1 ch2o 1 oh
1 ch2o 1 oh -> 1 ch2oh 1 o
1 ch3o 1 o2 -> 1 ch2o 1 ho2
1 ch2o 1 ho2 -> 1 ch3o 1 o2
1 ch3o 1 co -> 1 ch3 1 co2
1 ch3 1 co2 -> 1 ch3o 1 co
1 ch2oh 1 o2 -> 1 ch2o 1 ho2
1 ch2o 1 ho2 -> 1 ch2oh 1 o2
1 hcoh 1 oh -> 1 hco 1 h2o
1 hco 1 h2o -> 1 hcoh 1 oh
1 hcoh 1 h -> 1 ch2o 1 h
1 ch2o 1 h -> 1 hcoh 1 h
1 hcoh 1 o -> 1 co 1 oh 1 h
1 co 1 oh 1 h -> 1 hcoh 1 o
1 hcoh 1 o2 -> 1 co 1 oh 1 oh
1 co 1 oh 1 oh -> 1 hcoh 1 o2
1 hcoh 1 o2 -> 1 co2 1 h2o
1 co2 1 h2o -> 1 hcoh 1 o2
1 hcoh -> 1 ch2o
1 ch2o -> 1 hcoh
1 ch2 1 h -> 1 ch 1 h2
1 ch 1 h2 -> 1 ch2 1 h
1 ch2 1 oh -> 1 ch 1 h2o
1 ch 1 h2o -> 1 ch2 1 oh
1 ch2 1 oh -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch2 1 oh
1 ch2 1 co2 -> 1 ch2o 1 co
1 ch2o 1 co -> 1 ch2 1 co2
1 ch2 1 o -> 1 co 1 h 1 h
1 co 1 h 1 h -> 1 ch2 1 o
1 ch2 1 o -> 1 co 1 h2
1 co 1 h2 -> 1 ch2 1 o
1 ch2 1 o2 -> 1 ch2o 1 o
1 ch2o 1 o -> 1 ch2 1 o2
1 ch2 1 o2 -> 1 co2 1 h 1 h
1 co2 1 h 1 h -> 1 ch2 1 o2
1 ch2 1 o2 -> 1 co2 1 h2
1 co2 1 h2 -> 1 ch2 1 o2
1 ch2 1 o2 -> 1 co 1 h2o
1 co 1 h2o -> 1 ch2 1 o2
1 ch2 1 o2 -> 1 hco 1 oh
1 hco 1 oh -> 1 ch2 1 o2
1 ch2 1 ch3 -> 1 c2h4 1 h
1 c2h4 1 h -> 1 ch2 1 ch3
1 ch2 1 ch2 -> 1 c2h2 1 h 1 h
1 c2h2 1 h 1 h -> 1 ch2 1 ch2
1 ch2 1 hcco -> 1 c2h3 1 co
1 c2h3 1 co -> 1 ch2 1 hcco
1 ch2 1 c2h2 -> 1 h2ccch 1 h
1 h2ccch 1 h -> 1 ch2 1 c2h2
1 ch2(s) -> 1 ch2
1 ch2 -> 1 ch2(s)
1 ch2(s) 1 ch4 -> 1 ch3 1 ch3
1 ch3 1 ch3 -> 1 ch2(s) 1 ch4
1 ch2(s) 1 c2h6 -> 1 ch3 1 c2h5
1 ch3 1 c2h5 -> 1 ch2(s) 1 c2h6
1 ch2(s) 1 o2 -> 1 co 1 oh 1 h
1 co 1 oh 1 h -> 1 ch2(s) 1 o2
1 ch2(s) 1 h2 -> 1 ch3 1 h
1 ch3 1 h -> 1 ch2(s) 1 h2
1 ch2(s) 1 c2h2 -> 1 h2ccch 1 h
1 h2ccch 1 h -> 1 ch2(s) 1 c2h2
1 ch2(s) 1 c2h4 -> 1 ac3h5 1 h
1 ac3h5 1 h -> 1 ch2(s) 1 c2h4
1 ch2(s) 1 o -> 1 co 1 h 1 h
1 co 1 h 1 h -> 1 ch2(s) 1 o
1 ch2(s) 1 oh -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch2(s) 1 oh
1 ch2(s) 1 h -> 1 ch 1 h2
1 ch 1 h2 -> 1 ch2(s) 1 h
1 ch2(s) 1 co2 -> 1 ch2o 1 co
1 ch2o 1 co -> 1 ch2(s) 1 co2
1 ch2(s) 1 ch3 -> 1 c2h4 1 h
1 c2h4 1 h -> 1 ch2(s) 1 ch3
1 ch2(s) 1 ch2co -> 1 c2h4 1 co
```

```
1 c2h4 1 co -> 1 ch2(s) 1 ch2co
1 ch 1 o2 -> 1 hco 1 o
1 hco 1 o -> 1 ch 1 o2
1 ch 1 o -> 1 co 1 h
1 co 1 h -> 1 ch 1 o
1 ch 1 oh -> 1 hco 1 h
1 hco 1 h -> 1 ch 1 oh
1 ch 1 co2 -> 1 hco 1 co
1 hco 1 co -> 1 ch 1 co2
1 ch 1 h2o -> 1 ch2o 1 h
1 ch2o 1 h -> 1 ch 1 h2o
1 ch 1 ch2o -> 1 ch2co 1 h
1 ch2co 1 h -> 1 ch 1 ch2o
1 ch 1 c2h2 -> 1 c3h2 1 h
1 c3h2 1 h -> 1 ch 1 c2h2
1 ch 1 ch2 -> 1 c2h2 1 h
1 c2h2 1 h -> 1 ch 1 ch2
1 ch 1 ch3 -> 1 c2h3 1 h
1 c2h3 1 h -> 1 ch 1 ch3
1 ch 1 ch4 -> 1 c2h4 1 h
1 c2h4 1 h -> 1 ch 1 ch4
1 hcooh -> 1 co 1 h2o
1 co 1 h2o -> 1 hcooh
1 hcooh -> 1 co2 1 h2
1 co2 1 h2 -> 1 hcooh
1 hcooh 1 oh -> 1 co2 1 h2o 1 h
1 co2 1 h2o 1 h -> 1 hcooh 1 oh
1 hcooh 1 oh -> 1 co 1 h2o 1 oh
1 co 1 h2o 1 oh -> 1 hcooh 1 oh
1 hcooh 1 h -> 1 co2 1 h2 1 h
1 co2 1 h2 1 h -> 1 hcooh 1 h
1 hcooh 1 h -> 1 co 1 h2 1 oh
1 co 1 h2 1 oh -> 1 hcooh 1 h
1 hcooh 1 ch3 -> 1 ch4 1 co 1 oh
1 ch4 1 co 1 oh -> 1 hcooh 1 ch3
1 hcooh 1 ho2 -> 1 co 1 h2o2 1 oh
1 co 1 h2o2 1 oh -> 1 hcooh 1 ho2
1 hcooh 1 o -> 1 co 1 oh 1 oh
1 co 1 oh 1 oh -> 1 hcooh 1 o
1 ch2o 1 oh -> 1 hco 1 h2o
1 hco 1 h2o -> 1 ch2o 1 oh
1 ch2o 1 h -> 1 hco 1 h2
1 hco 1 h2 -> 1 ch2o 1 h
1 ch2o -> 1 hco 1 h
1 hco 1 h -> 1 ch2o
1 ch2o 1 o -> 1 hco 1 oh
1 hco 1 oh -> 1 ch2o 1 o
1 hco 1 o2 -> 1 co 1 ho2
1 co 1 ho2 -> 1 hco 1 o2
1 hco -> 1 h 1 co
1 h 1 co -> 1 hco
1 hco 1 oh -> 1 h2o 1 co
1 h2o 1 co -> 1 hco 1 oh
1 hco 1 h -> 1 co 1 h2
1 co 1 h2 -> 1 hco 1 h
1 hco 1 o -> 1 co 1 oh
1 co 1 oh -> 1 hco 1 o
1 hco 1 o -> 1 co2 1 h
1 co2 1 h -> 1 hco 1 o
1 co 1 oh -> 1 co2 1 h
1 co2 1 h -> 1 co 1 oh
1 co 1 o -> 1 co2
1 co2 -> 1 co 1 o
1 co 1 o2 -> 1 co2 1 o
1 co2 1 o -> 1 co 1 o2
1 co 1 ho2 -> 1 co2 1 oh
1 co2 1 oh -> 1 co 1 ho2
1 c2h5oh -> 1 ch3 1 ch2oh
1 ch3 1 ch2oh -> 1 c2h5oh
1 c2h5oh -> 1 c2h5 1 oh
1 c2h5 1 oh -> 1 c2h5oh
1 c2h5oh -> 1 c2h4 1 h2o
1 c2h4 1 h2o -> 1 c2h5oh
1 c2h5oh -> 1 ch3hco 1 h2
1 ch3hco 1 h2 -> 1 c2h5oh
1 c2h5oh 1 oh -> 1 c2h4oh 1 h2o
1 c2h4oh 1 h2o -> 1 c2h5oh 1 oh
1 c2h5oh 1 oh -> 1 ch3choh 1 h2o
1 ch3choh 1 h2o -> 1 c2h5oh 1 oh
1 c2h5oh 1 oh -> 1 ch3ch2o 1 h2o
1 ch3ch2o 1 h2o -> 1 c2h5oh 1 oh
1 c2h5oh 1 h -> 1 c2h4oh 1 h2
1 c2h4oh 1 h2 -> 1 c2h5oh 1 h
1 c2h5oh 1 h -> 1 ch3choh 1 h2
1 ch3choh 1 h2 -> 1 c2h5oh 1 h
1 c2h5oh 1 h -> 1 ch3ch2o 1 h2
1 ch3ch2o 1 h2 -> 1 c2h5oh 1 h
1 c2h5oh 1 o -> 1 c2h4oh 1 oh
1 c2h4oh 1 oh -> 1 c2h5oh 1 o
1 c2h5oh 1 o -> 1 ch3choh 1 oh
1 ch3choh 1 oh -> 1 c2h5oh 1 o
1 c2h5oh 1 o -> 1 ch3ch2o 1 oh
1 ch3ch2o 1 oh -> 1 c2h5oh 1 o
1 c2h5oh 1 ch3 -> 1 c2h4oh 1 ch4
1 c2h4oh 1 ch4 -> 1 c2h5oh 1 ch3
1 c2h5oh 1 ch3 -> 1 ch3choh 1 ch4
1 ch3choh 1 ch4 -> 1 c2h5oh 1 ch3
1 c2h5oh 1 ch3 -> 1 ch3ch2o 1 ch4
1 ch3ch2o 1 ch4 -> 1 c2h5oh 1 ch3
1 c2h5oh 1 ho2 -> 1 ch3choh 1 h2o2
1 ch3choh 1 h2o2 -> 1 c2h5oh 1 ho2
1 c2h5oh 1 ho2 -> 1 c2h4oh 1 h2o2
1 c2h4oh 1 h2o2 -> 1 c2h5oh 1 ho2
1 c2h5oh 1 ho2 -> 1 ch3ch2o 1 h2o2
1 ch3ch2o 1 h2o2 -> 1 c2h5oh 1 ho2
1 ch3ch2o -> 1 ch3hco 1 h
1 ch3hco 1 h -> 1 ch3ch2o
1 ch3ch2o -> 1 ch3 1 ch2o
1 ch3 1 ch2o -> 1 ch3ch2o
1 ch3ch2o 1 o2 -> 1 ch3hco 1 ho2
1 ch3hco 1 ho2 -> 1 ch3ch2o 1 o2
1 ch3ch2o 1 co -> 1 c2h5 1 co2
1 c2h5 1 co2 -> 1 ch3ch2o 1 co
1 ch3ch2o 1 h -> 1 ch3 1 ch2oh
1 ch3 1 ch2oh -> 1 ch3ch2o 1 h
1 ch3ch2o 1 h -> 1 c2h4 1 h2o
1 c2h4 1 h2o -> 1 ch3ch2o 1 h
1 ch3ch2o 1 oh -> 1 ch3hco 1 h2o
1 ch3hco 1 h2o -> 1 ch3ch2o 1 oh
1 ch3choh 1 o2 -> 1 ch3hco 1 ho2
1 ch3hco 1 ho2 -> 1 ch3choh 1 o2
1 ch3choh 1 ch3 -> 1 c3h6 1 h2o
1 c3h6 1 h2o -> 1 ch3choh 1 ch3
1 ch3choh 1 o -> 1 ch3hco 1 oh
1 ch3hco 1 oh -> 1 ch3choh 1 o
1 ch3choh 1 h -> 1 c2h4 1 h2o
1 c2h4 1 h2o -> 1 ch3choh 1 h
1 ch3choh 1 h -> 1 ch3 1 ch2oh
1 ch3 1 ch2oh -> 1 ch3choh 1 h
1 ch3choh 1 ho2 -> 1 ch3hco 1 oh 1 oh
1 ch3hco 1 oh 1 oh -> 1 ch3choh 1 ho2
1 ch3choh 1 oh -> 1 ch3hco 1 h2o
1 ch3hco 1 h2o -> 1 ch3choh 1 oh
1 ch3choh -> 1 ch3hco 1 h
1 ch3hco 1 h -> 1 ch3choh
1 ch3hco 1 oh -> 1 ch3co 1 h2o
1 ch3co 1 h2o -> 1 ch3hco 1 oh
1 ch3hco 1 oh -> 1 ch2hco 1 h2o
1 ch2hco 1 h2o -> 1 ch3hco 1 oh
1 ch3hco 1 oh -> 1 ch3 1 hcooh
1 ch3 1 hcooh -> 1 ch3hco 1 oh
1 ch3hco 1 o -> 1 ch3co 1 oh
1 ch3co 1 oh -> 1 ch3hco 1 o
1 ch3hco 1 o -> 1 ch2hco 1 oh
1 ch2hco 1 oh -> 1 ch3hco 1 o
1 ch3hco 1 h -> 1 ch3co 1 h2
```

```
1 ch3co 1 h2 -> 1 ch3hco 1 h
1 ch3hco 1 h -> 1 ch2hco 1 h2
1 ch2hco 1 h2 -> 1 ch3hco 1 h
1 ch3hco 1 ch3 -> 1 ch3co 1 ch4
1 ch3co 1 ch4 -> 1 ch3hco 1 ch3
1 ch3hco 1 ch3 -> 1 ch2hco 1 ch4
1 ch2hco 1 ch4 -> 1 ch3hco 1 ch3
1 ch3hco 1 ho2 -> 1 ch3co 1 h2o2
1 ch3co 1 h2o2 -> 1 ch3hco 1 ho2
1 ch3hco 1 ho2 -> 1 ch2hco 1 h2o2
1 ch2hco 1 h2o2 -> 1 ch3hco 1 ho2
1 ch3hco 1 o2 -> 1 ch3co 1 ho2
1 ch3co 1 ho2 -> 1 ch3hco 1 o2
1 c2h6 1 ch3 -> 1 c2h5 1 ch4
1 c2h5 1 ch4 -> 1 c2h6 1 ch3
1 c2h6 1 h -> 1 c2h5 1 h2
1 c2h5 1 h2 -> 1 c2h6 1 h
1 c2h6 1 o -> 1 c2h5 1 oh
1 c2h5 1 oh -> 1 c2h6 1 o
1 c2h6 1 oh -> 1 c2h5 1 h2o
1 c2h5 1 h2o -> 1 c2h6 1 oh
1 c2h5 1 h -> 1 c2h4 1 h2
1 c2h4 1 h2 -> 1 c2h5 1 h
1 c2h5 1 h -> 1 ch3 1 ch3
1 ch3 1 ch3 -> 1 c2h5 1 h
1 c2h5 1 h -> 1 c2h6
1 c2h6 -> 1 c2h5 1 h
1 c2h5 1 oh -> 1 c2h4 1 h2o
1 c2h4 1 h2o -> 1 c2h5 1 oh
1 c2h5 1 o -> 1 ch3 1 ch2o
1 ch3 1 ch2o -> 1 c2h5 1 o
1 c2h5 1 ho2 -> 1 c2h6 1 o2
1 c2h6 1 o2 -> 1 c2h5 1 ho2
1 c2h5 1 ho2 -> 1 ch3ch2o 1 oh
1 ch3ch2o 1 oh -> 1 c2h5 1 ho2
1 c2h5 1 o2 -> 1 c2h4 1 ho2
1 c2h4 1 ho2 -> 1 c2h5 1 o2
1 c2h5 1 o2 -> 1 ch3hco 1 oh
1 ch3hco 1 oh -> 1 c2h5 1 o2
1 c2h4 1 oh -> 1 c2h4oh
1 c2h4oh -> 1 c2h4 1 oh
1 c2h4oh 1 o2 -> 1 hoc2h4o2
1 hoc2h4o2 -> 1 c2h4oh 1 o2
1 hoc2h4o2 -> 1 ch2o 1 ch2o 1 oh
1 ch2o 1 ch2o 1 oh -> 1 hoc2h4o2
1 c2h4 1 oh -> 1 c2h3 1 h2o
1 c2h3 1 h2o -> 1 c2h4 1 oh
1 c2h4 1 o -> 1 ch3 1 hco
1 ch3 1 hco -> 1 c2h4 1 o
1 c2h4 1 o -> 1 ch2hco 1 h
1 ch2hco 1 h -> 1 c2h4 1 o
1 c2h4 1 ch3 -> 1 c2h3 1 ch4
1 c2h3 1 ch4 -> 1 c2h4 1 ch3
1 c2h4 1 h -> 1 c2h3 1 h2
1 c2h3 1 h2 -> 1 c2h4 1 h
1 c2h4 -> 1 c2h2 1 h2
1 c2h2 1 h2 -> 1 c2h4
1 c2h3 1 h -> 1 c2h4
1 c2h4 -> 1 c2h3 1 h
1 c2h3 1 h -> 1 c2h2 1 h2
1 c2h2 1 h2 -> 1 c2h3 1 h
1 c2h3 1 o -> 1 ch2co 1 h
1 ch2co 1 h -> 1 c2h3 1 o
1 c2h3 1 o2 -> 1 ch2o 1 hco
1 ch2o 1 hco -> 1 c2h3 1 o2
1 c2h3 1 o2 -> 1 ch2hco 1 o
1 ch2hco 1 o -> 1 c2h3 1 o2
1 c2h3 1 o2 -> 1 c2h2 1 ho2
1 c2h2 1 ho2 -> 1 c2h3 1 o2
1 c2h3 1 oh -> 1 c2h2 1 h2o
1 c2h2 1 h2o -> 1 c2h3 1 oh
1 c2h3 1 c2h -> 1 c2h2 1 c2h2
1 c2h2 1 c2h2 -> 1 c2h3 1 c2h
1 c2h3 1 ch -> 1 ch2 1 c2h2
1 ch2 1 c2h2 -> 1 c2h3 1 ch
1 c2h3 1 ch3 -> 1 ac3h5 1 h
1 ac3h5 1 h -> 1 c2h3 1 ch3
1 c2h3 1 ch3 -> 1 c3h6
1 c3h6 -> 1 c2h3 1 ch3
1 c2h3 1 ch3 -> 1 c2h2 1 ch4
1 c2h2 1 ch4 -> 1 c2h3 1 ch3
1 c2h2 1 oh -> 1 c2h 1 h2o
1 c2h 1 h2o -> 1 c2h2 1 oh
1 c2h2 1 oh -> 1 hccoh 1 h
1 hccoh 1 h -> 1 c2h2 1 oh
1 c2h2 1 oh -> 1 ch2co 1 h
1 ch2co 1 h -> 1 c2h2 1 oh
1 c2h2 1 oh -> 1 ch3 1 co
1 ch3 1 co -> 1 c2h2 1 oh
1 hccoh 1 h -> 1 ch2co 1 h
1 ch2co 1 h -> 1 hccoh 1 h
1 c2h2 1 o -> 1 ch2 1 co
1 ch2 1 co -> 1 c2h2 1 o
1 c2h2 1 o -> 1 hcco 1 h
1 hcco 1 h -> 1 c2h2 1 o
1 c2h2 1 o -> 1 c2h 1 oh
1 c2h 1 oh -> 1 c2h2 1 o
1 c2h2 1 ch3 -> 1 c2h 1 ch4
1 c2h 1 ch4 -> 1 c2h2 1 ch3
1 c2h2 1 o2 -> 1 hcco 1 oh
1 hcco 1 oh -> 1 c2h2 1 o2
1 c2h2 -> 1 c2h 1 h
1 c2h 1 h -> 1 c2h2
1 ch2hco 1 h -> 1 ch3 1 hco
1 ch3 1 hco -> 1 ch2hco 1 h
1 ch2hco 1 h -> 1 ch2co 1 h2
1 ch2co 1 h2 -> 1 ch2hco 1 h
1 ch2hco 1 o -> 1 ch2o 1 hco
1 ch2o 1 hco -> 1 ch2hco 1 o
1 ch2hco 1 oh -> 1 ch2co 1 h2o
1 ch2co 1 h2o -> 1 ch2hco 1 oh
1 ch2hco 1 o2 -> 1 ch2o 1 co 1 oh
1 ch2o 1 co 1 oh -> 1 ch2hco 1 o2
1 ch2hco 1 ch3 -> 1 c2h5 1 co 1 h
1 c2h5 1 co 1 h -> 1 ch2hco 1 ch3
1 ch2hco 1 ho2 -> 1 ch2o 1 hco 1 oh
1 ch2o 1 hco 1 oh -> 1 ch2hco 1 ho2
1 ch2hco 1 ho2 -> 1 ch3hco 1 o2
1 ch3hco 1 o2 -> 1 ch2hco 1 ho2
1 ch2hco -> 1 ch3 1 co
1 ch3 1 co -> 1 ch2hco
1 ch2hco -> 1 ch2co 1 h
1 ch2co 1 h -> 1 ch2hco
1 chocho -> 1 ch2o 1 co
1 ch2o 1 co -> 1 chocho
1 chocho -> 1 co 1 co 1 h2
1 co 1 co 1 h2 -> 1 chocho
1 chocho 1 oh -> 1 hco 1 co 1 h2o
1 hco 1 co 1 h2o -> 1 chocho 1 oh
1 chocho 1 o -> 1 hco 1 co 1 oh
1 hco 1 co 1 oh -> 1 chocho 1 o
1 chocho 1 h -> 1 ch2o 1 hco
1 ch2o 1 hco -> 1 chocho 1 h
1 chocho 1 ho2 -> 1 hco 1 co 1 h2o2
1 hco 1 co 1 h2o2 -> 1 chocho 1 ho2
1 chocho 1 ch3 -> 1 hco 1 co 1 ch4
1 hco 1 co 1 ch4 -> 1 chocho 1 ch3
1 chocho 1 o2 -> 1 hco 1 co 1 ho2
1 hco 1 co 1 ho2 -> 1 chocho 1 o2
1 ch3co -> 1 ch3 1 co
1 ch3 1 co -> 1 ch3co
1 ch2co 1 o -> 1 co2 1 ch2
1 co2 1 ch2 -> 1 ch2co 1 o
1 ch2co 1 h -> 1 ch3 1 co
1 ch3 1 co -> 1 ch2co 1 h
1 ch2co 1 h -> 1 hcco 1 h2
```

```
1 hcco 1 h2 -> 1 ch2co 1 h
1 ch2co 1 o -> 1 hcco 1 oh
1 hcco 1 oh -> 1 ch2co 1 o
1 ch2co 1 oh -> 1 hcco 1 h2o
1 hcco 1 h2o -> 1 ch2co 1 oh
1 ch2co 1 oh -> 1 ch2oh 1 co
1 ch2oh 1 co -> 1 ch2co 1 oh
1 ch2co -> 1 ch2 1 co
1 ch2 1 co -> 1 ch2co
1 c2h 1 h2 -> 1 c2h2 1 h
1 c2h2 1 h -> 1 c2h 1 h2
1 c2h 1 o -> 1 ch 1 co
1 ch 1 co -> 1 c2h 1 o
1 c2h 1 oh -> 1 hcco 1 h
1 hcco 1 h -> 1 c2h 1 oh
1 c2h 1 o2 -> 1 co 1 co 1 h
1 co 1 co 1 h -> 1 c2h 1 o2
1 hcco 1 c2h2 -> 1 h2ccch 1 co
1 h2ccch 1 co -> 1 hcco 1 c2h2
1 hcco 1 h -> 1 ch2(s) 1 co
1 ch2(s) 1 co -> 1 hcco 1 h
1 hcco 1 o -> 1 h 1 co 1 co
1 h 1 co 1 co -> 1 hcco 1 o
1 hcco 1 o -> 1 ch 1 co2
1 ch 1 co2 -> 1 hcco 1 o
1 hcco 1 o2 -> 1 hco 1 co 1 o
1 hco 1 co 1 o -> 1 hcco 1 o2
1 hcco 1 o2 -> 1 co2 1 hco
1 co2 1 hco -> 1 hcco 1 o2
1 hcco 1 ch -> 1 c2h2 1 co
1 c2h2 1 co -> 1 hcco 1 ch
1 hcco 1 hcco -> 1 c2h2 1 co 1 co
1 c2h2 1 co 1 co -> 1 hcco 1 hcco
1 hcco 1 oh -> 1 c2o 1 h2o
1 c2o 1 h2o -> 1 hcco 1 oh
1 c2o 1 h -> 1 ch 1 co
1 ch 1 co -> 1 c2o 1 h
1 c2o 1 o -> 1 co 1 co
1 co 1 co -> 1 c2o 1 o
1 c2o 1 oh -> 1 co 1 co 1 h
1 co 1 co 1 h -> 1 c2o 1 oh
1 c2o 1 o2 -> 1 co 1 co 1 o
1 co 1 co 1 o -> 1 c2o 1 o2
1 c3h8 -> 1 c2h5 1 ch3
1 c2h5 1 ch3 -> 1 c3h8
1 ic3h7 1 ho2 -> 1 c3h8 1 o2
1 c3h8 1 o2 -> 1 ic3h7 1 ho2
1 nc3h7 1 ho2 -> 1 c3h8 1 o2
1 c3h8 1 o2 -> 1 nc3h7 1 ho2
1 c3h8 1 ho2 -> 1 nc3h7 1 h2o2
1 nc3h7 1 h2o2 -> 1 c3h8 1 ho2
1 c3h8 1 ho2 -> 1 ic3h7 1 h2o2
1 ic3h7 1 h2o2 -> 1 c3h8 1 ho2
1 c3h8 1 oh -> 1 nc3h7 1 h2o
1 nc3h7 1 h2o -> 1 c3h8 1 oh
1 c3h8 1 oh -> 1 ic3h7 1 h2o
1 ic3h7 1 h2o -> 1 c3h8 1 oh
1 c3h8 1 o -> 1 nc3h7 1 oh
1 nc3h7 1 oh -> 1 c3h8 1 o
1 c3h8 1 o -> 1 ic3h7 1 oh
1 ic3h7 1 oh -> 1 c3h8 1 o
1 c3h8 1 h -> 1 ic3h7 1 h2
1 ic3h7 1 h2 -> 1 c3h8 1 h
1 c3h8 1 h -> 1 nc3h7 1 h2
1 nc3h7 1 h2 -> 1 c3h8 1 h
1 c3h8 1 ch3 -> 1 nc3h7 1 ch4
1 nc3h7 1 ch4 -> 1 c3h8 1 ch3
1 c3h8 1 ch3 -> 1 ic3h7 1 ch4
1 ic3h7 1 ch4 -> 1 c3h8 1 ch3
1 c3h8 1 c2h3 -> 1 ic3h7 1 c2h4
1 ic3h7 1 c2h4 -> 1 c3h8 1 c2h3
1 c3h8 1 c2h3 -> 1 nc3h7 1 c2h4
1 nc3h7 1 c2h4 -> 1 c3h8 1 c2h3
1 c3h8 1 c2h5 -> 1 ic3h7 1 c2h6
1 ic3h7 1 c2h6 -> 1 c3h8 1 c2h5
1 c3h8 1 c2h5 -> 1 nc3h7 1 c2h6
1 nc3h7 1 c2h6 -> 1 c3h8 1 c2h5
1 c3h8 1 ac3h5 -> 1 c3h6 1 nc3h7
1 c3h6 1 nc3h7 -> 1 c3h8 1 ac3h5
1 c3h8 1 ac3h5 -> 1 c3h6 1 ic3h7
1 c3h6 1 ic3h7 -> 1 c3h8 1 ac3h5
1 nc3h7 -> 1 c2h4 1 ch3
1 c2h4 1 ch3 -> 1 nc3h7
1 c3h6 1 h -> 1 ic3h7
1 ic3h7 -> 1 c3h6 1 h
1 ic3h7 1 o2 -> 1 c3h6 1 ho2
1 c3h6 1 ho2 -> 1 ic3h7 1 o2
1 nc3h7 1 o2 -> 1 c3h6 1 ho2
1 c3h6 1 ho2 -> 1 nc3h7 1 o2
1 ic3h7 1 h -> 1 c2h5 1 ch3
1 c2h5 1 ch3 -> 1 ic3h7 1 h
1 nc3h7 1 h -> 1 c2h5 1 ch3
1 c2h5 1 ch3 -> 1 nc3h7 1 h
1 c3h6 -> 1 c2h2 1 ch4
1 c2h2 1 ch4 -> 1 c3h6
1 c3h6 -> 1 ac3h4 1 h2
1 ac3h4 1 h2 -> 1 c3h6
1 pc3h5 1 h -> 1 c3h6
1 c3h6 -> 1 pc3h5 1 h
1 sc3h5 1 h -> 1 c3h6
1 c3h6 -> 1 sc3h5 1 h
1 c3h6 1 ho2 -> 1 ac3h5 1 h2o2
1 ac3h5 1 h2o2 -> 1 c3h6 1 ho2
1 c3h6 1 oh -> 1 ac3h5 1 h2o
1 ac3h5 1 h2o -> 1 c3h6 1 oh
1 c3h6 1 oh -> 1 sc3h5 1 h2o
1 sc3h5 1 h2o -> 1 c3h6 1 oh
1 c3h6 1 oh -> 1 pc3h5 1 h2o
1 pc3h5 1 h2o -> 1 c3h6 1 oh
1 c3h6 1 o -> 1 ch3chco 1 h 1 h
1 ch3chco 1 h 1 h -> 1 c3h6 1 o
1 c3h6 1 o -> 1 c2h5 1 hco
1 c2h5 1 hco -> 1 c3h6 1 o
1 c3h6 1 o -> 1 ac3h5 1 oh
1 ac3h5 1 oh -> 1 c3h6 1 o
1 c3h6 1 o -> 1 pc3h5 1 oh
1 pc3h5 1 oh -> 1 c3h6 1 o
1 c3h6 1 o -> 1 sc3h5 1 oh
1 sc3h5 1 oh -> 1 c3h6 1 o
1 c3h6 1 h -> 1 c2h4 1 ch3
1 c2h4 1 ch3 -> 1 c3h6 1 h
1 c3h6 1 h -> 1 ac3h5 1 h2
1 ac3h5 1 h2 -> 1 c3h6 1 h
1 c3h6 1 h -> 1 sc3h5 1 h2
1 sc3h5 1 h2 -> 1 c3h6 1 h
1 c3h6 1 h -> 1 pc3h5 1 h2
1 pc3h5 1 h2 -> 1 c3h6 1 h
1 ac3h5 1 ho2 -> 1 c3h6 1 o2
1 c3h6 1 o2 -> 1 ac3h5 1 ho2
1 c3h6 1 ch3 -> 1 ac3h5 1 ch4
1 ac3h5 1 ch4 -> 1 c3h6 1 ch3
1 c3h6 1 ch3 -> 1 sc3h5 1 ch4
1 sc3h5 1 ch4 -> 1 c3h6 1 ch3
1 c3h6 1 ch3 -> 1 pc3h5 1 ch4
1 pc3h5 1 ch4 -> 1 c3h6 1 ch3
1 c3h6 1 hco -> 1 ac3h5 1 ch2o
1 ac3h5 1 ch2o -> 1 c3h6 1 hco
1 ch3chco 1 oh -> 1 ch2chco 1 h2o
1 ch2chco 1 h2o -> 1 ch3chco 1 oh
1 ch3chco 1 o -> 1 ch2chco 1 oh
1 ch2chco 1 oh -> 1 ch3chco 1 o
1 ch3chco 1 h -> 1 ch2chco 1 h2
1 ch2chco 1 h2 -> 1 ch3chco 1 h
1 ch3chco 1 h -> 1 c2h5 1 co
1 c2h5 1 co -> 1 ch3chco 1 h
1 ch3chco 1 o -> 1 ch3 1 hco 1 co
```

```
1 ch3 1 hco 1 co -> 1 ch3chco 1 o
1 ch2chcho 1 oh -> 1 ch2chco 1 h2o
1 ch2chco 1 h2o -> 1 ch2chcho 1 oh
1 ch2chcho 1 o -> 1 ch2chco 1 oh
1 ch2chco 1 oh -> 1 ch2chcho 1 o
1 ch2chcho 1 o -> 1 ch2co 1 hco 1 h
1 ch2co 1 hco 1 h -> 1 ch2chcho 1 o
1 ch2chcho 1 h -> 1 ch2chco 1 h2
1 ch2chco 1 h2 -> 1 ch2chcho 1 h
1 ch2chcho 1 h -> 1 c2h4 1 hco
1 c2h4 1 hco -> 1 ch2chcho 1 h
1 ch2chcho 1 o2 -> 1 ch2chco 1 ho2
1 ch2chco 1 ho2 -> 1 ch2chcho 1 o2
1 ch2chco -> 1 c2h3 1 co
1 c2h3 1 co -> 1 ch2chco
1 ch2chco 1 o -> 1 c2h3 1 co2
1 c2h3 1 co2 -> 1 ch2chco 1 o
1 ac3h5 1 o2 -> 1 ch2chcho 1 oh
1 ch2chcho 1 oh -> 1 ac3h5 1 o2
1 ac3h5 1 o2 -> 1 ac3h4 1 ho2
1 ac3h4 1 ho2 -> 1 ac3h5 1 o2
1 ac3h5 1 o2 -> 1 ch2hco 1 ch2o
1 ch2hco 1 ch2o -> 1 ac3h5 1 o2
1 ac3h5 1 o2 -> 1 c2h2 1 ch2o 1 oh
1 c2h2 1 ch2o 1 oh -> 1 ac3h5 1 o2
1 ac3h5 1 ho2 -> 1 ch2chch2o 1 oh
1 ch2chch2o 1 oh -> 1 ac3h5 1 ho2
1 ch2chch2o 1 o2 -> 1 ch2chcho 1 ho2
1 ch2chcho 1 ho2 -> 1 ch2chch2o 1 o2
1 ch2chch2o 1 co -> 1 ac3h5 1 co2
1 ac3h5 1 co2 -> 1 ch2chch2o 1 co
1 ch2chcho 1 h -> 1 ch2chch2o
1 ch2chch2o -> 1 ch2chcho 1 h
1 ac3h5 1 oh -> 1 ac3h4 1 h2o
1 ac3h4 1 h2o -> 1 ac3h5 1 oh
1 ac3h5 1 h -> 1 ac3h4 1 h2
1 ac3h4 1 h2 -> 1 ac3h5 1 h
1 ac3h5 1 h -> 1 c3h6
1 c3h6 -> 1 ac3h5 1 h
1 ac3h5 1 o -> 1 ch2chcho 1 h
1 ch2chcho 1 h -> 1 ac3h5 1 o
1 ac3h5 1 ch3 -> 1 ac3h4 1 ch4
1 ac3h4 1 ch4 -> 1 ac3h5 1 ch3
1 pc3h5 1 o2 -> 1 ch3hco 1 hco
1 ch3hco 1 hco -> 1 pc3h5 1 o2
1 pc3h5 1 o2 -> 1 ch3chco 1 h 1 o
1 ch3chco 1 h 1 o -> 1 pc3h5 1 o2
1 pc3h5 1 o -> 1 ch3chco 1 h
1 ch3chco 1 h -> 1 pc3h5 1 o
1 pc3h5 1 h -> 1 pc3h4 1 h2
1 pc3h4 1 h2 -> 1 pc3h5 1 h
1 pc3h5 1 oh -> 1 pc3h4 1 h2o
1 pc3h4 1 h2o -> 1 pc3h5 1 oh
1 pc3h5 1 h -> 1 ac3h5 1 h
1 ac3h5 1 h -> 1 pc3h5 1 h
1 sc3h5 1 h -> 1 ac3h5 1 h
1 ac3h5 1 h -> 1 sc3h5 1 h
1 sc3h5 1 o2 -> 1 ch3co 1 ch2o
1 ch3co 1 ch2o -> 1 sc3h5 1 o2
1 sc3h5 1 o -> 1 ch2co 1 ch3
1 ch2co 1 ch3 -> 1 sc3h5 1 o
1 sc3h5 1 h -> 1 pc3h4 1 h2
1 pc3h4 1 h2 -> 1 sc3h5 1 h
1 sc3h5 1 oh -> 1 pc3h4 1 h2o
1 pc3h4 1 h2o -> 1 sc3h5 1 oh
1 ac3h4 1 h -> 1 h2ccch 1 h2
1 h2ccch 1 h2 -> 1 ac3h4 1 h
1 ac3h4 1 o -> 1 c2h4 1 co
1 c2h4 1 co -> 1 ac3h4 1 o
1 ac3h4 1 oh -> 1 h2ccch 1 h2o
1 h2ccch 1 h2o -> 1 ac3h4 1 oh
1 ac3h4 1 ch3 -> 1 h2ccch 1 ch4
1 h2ccch 1 ch4 -> 1 ac3h4 1 ch3
1 ac3h4 -> 1 pc3h4
1 pc3h4 -> 1 ac3h4
1 pc3h4 1 h -> 1 h2ccch 1 h2
1 h2ccch 1 h2 -> 1 pc3h4 1 h
1 pc3h4 1 o -> 1 c2h4 1 co
1 c2h4 1 co -> 1 pc3h4 1 o
1 pc3h4 1 oh -> 1 h2ccch 1 h2o
1 h2ccch 1 h2o -> 1 pc3h4 1 oh
1 pc3h4 1 ch3 -> 1 h2ccch 1 ch4
1 h2ccch 1 ch4 -> 1 pc3h4 1 ch3
1 pc3h4 1 h -> 1 ch3 1 c2h2
1 ch3 1 c2h2 -> 1 pc3h4 1 h
1 pc3h4 1 h -> 1 sc3h5
1 sc3h5 -> 1 pc3h4 1 h
1 ac3h4 1 h -> 1 ac3h5
1 ac3h5 -> 1 ac3h4 1 h
1 ac3h4 1 h -> 1 sc3h5
1 sc3h5 -> 1 ac3h4 1 h
1 h2ccch 1 o2 -> 1 ch2co 1 hco
1 ch2co 1 hco -> 1 h2ccch 1 o2
1 h2ccch 1 o -> 1 ch2o 1 c2h
1 ch2o 1 c2h -> 1 h2ccch 1 o
1 h2ccch 1 h -> 1 c3h2 1 h2
1 c3h2 1 h2 -> 1 h2ccch 1 h
1 h2ccch 1 oh -> 1 c3h2 1 h2o
1 c3h2 1 h2o -> 1 h2ccch 1 oh
1 h2ccch 1 ch3 -> 1 c3h2 1 ch4
1 c3h2 1 ch4 -> 1 h2ccch 1 ch3
1 h2ccch 1 h -> 1 ac3h4
1 ac3h4 -> 1 h2ccch 1 h
1 h2ccch 1 h -> 1 pc3h4
1 pc3h4 -> 1 h2ccch 1 h
1 c3h2 1 o2 -> 1 hcco 1 co 1 h
1 hcco 1 co 1 h -> 1 c3h2 1 o2
1 c3h2 1 o -> 1 c2h2 1 co
1 c2h2 1 co -> 1 c3h2 1 o
1 c3h2 1 oh -> 1 c2h2 1 hco
1 c2h2 1 hco -> 1 c3h2 1 oh
```

E 2.3 Hydrogen

```
# Number of Components:
10
# Components:
h
h2
o
o2
oh
h2o
n2
ho2
h2o2
ar
# Number of Reactions:
38
# Reactions:
1 h 1 o2 -> 1 o 1 oh
1 o 1 oh -> 1 h 1 o2
1 o 1 h2 -> 1 h 1 oh
1 h 1 oh -> 1 o 1 h2
1 oh 1 h2 -> 1 h 1 h2o
1 h 1 h2o -> 1 oh 1 h2
1 o 1 h2o -> 1 oh 1 oh
1 oh 1 oh -> 1 o 1 h2o
1 h2 -> 1 h 1 h
1 h 1 h -> 1 h2
1 o2 -> 1 o 1 o
1 o 1 o -> 1 o2
1 oh -> 1 o 1 h
1 o 1 h -> 1 oh
1 h2o -> 1 h 1 oh
```

```
1 h 1 oh -> 1 h2o
1 h 1 o2 -> 1 ho2
1 ho2 -> 1 h 1 o2
1 ho2 1 h -> 1 h2 1 o2
1 h2 1 o2 -> 1 ho2 1 h
1 ho2 1 h -> 1 oh 1 oh
1 oh 1 oh -> 1 ho2 1 h
1 ho2 1 o -> 1 oh 1 o2
1 oh 1 o2 -> 1 ho2 1 o
1 ho2 1 oh -> 1 h2o 1 o2
1 h2o 1 o2 -> 1 ho2 1 oh
1 h2o2 1 o2 -> 1 ho2 1 ho2
1 ho2 1 ho2 -> 1 h2o2 1 o2
1 h2o2 -> 1 oh 1 oh
1 oh 1 oh -> 1 h2o2
1 h2o2 1 h -> 1 h2o 1 oh
1 h2o 1 oh -> 1 h2o2 1 h
1 h2o2 1 h -> 1 h2 1 ho2
1 h2 1 ho2 -> 1 h2o2 1 h
1 h2o2 1 o -> 1 oh 1 ho2
1 oh 1 ho2 -> 1 h2o2 1 o
1 h2o2 1 oh -> 1 h2o 1 ho2
1 h2o 1 ho2 -> 1 h2o2 1 oh
```

E 2.4 Methane

```
# Number of Components:
37
# Components:
H2
CH4
C2H2
C2H4
C2H6
C3H4
C3H6
C4H2
O2
H2O
H2O2
CO
CO2
CH2O
CH2CO
C
H
CH
CH2
CH2(S)
CH3
C2H
C2H3
C2H5
C3H2
H2CCCH
H2CCCCH
O
OH
HO2
HCO
CH3O
CH2OH
HCCO
CH2HCO
N2
AR
# Number of Reactions:
340
# Reactions:
1 H2 1 CH2(S) -> 1 CH3 1 H
1 CH3 1 H -> 1 H2 1 CH2(S)
1 H2 1 O -> 1 OH 1 H
1 OH 1 H -> 1 H2 1 O
1 H2O 1 H -> 1 H2 1 OH
1 H2 1 OH -> 1 H2O 1 H
1 CH4 1 O2 -> 1 CH3 1 HO2
1 CH3 1 HO2 -> 1 CH4 1 O2
1 CH4 1 C -> 1 CH 1 CH3
1 CH 1 CH3 -> 1 CH4 1 C
1 CH4 1 H -> 1 CH3 1 H2
1 CH3 1 H2 -> 1 CH4 1 H
1 CH4 1 CH -> 1 C2H4 1 H
1 C2H4 1 H -> 1 CH4 1 CH
1 CH4 1 CH2 -> 1 CH3 1 CH3
1 CH3 1 CH3 -> 1 CH4 1 CH2
1 CH4 1 CH2(S) -> 1 CH3 1 CH3
1 CH3 1 CH3 -> 1 CH4 1 CH2(S)
1 CH4 1 C2H -> 1 CH3 1 C2H2
1 CH3 1 C2H2 -> 1 CH4 1 C2H
1 CH4 1 O -> 1 CH3 1 OH
1 CH3 1 OH -> 1 CH4 1 O
1 CH4 1 OH -> 1 CH3 1 H2O
1 CH3 1 H2O -> 1 CH4 1 OH
1 CH4 1 HO2 -> 1 CH3 1 H2O2
1 CH3 1 H2O2 -> 1 CH4 1 HO2
1 C2H2 1 C2H2 -> 1 H2CCCCH
1 H2CCCCH -> 1 C2H2 1 C2H2
1 C2H2 1 O2 -> 1 C2H 1 HO2
1 C2H 1 HO2 -> 1 C2H2 1 O2
1 H2 1 C2H -> 1 C2H2 1 H
1 C2H2 1 H -> 1 H2 1 C2H
1 C2H2 1 CH -> 1 C2H 1 CH2
1 C2H 1 CH2 -> 1 C2H2 1 CH
1 C2H2 1 CH2 -> 1 C3H4
1 C3H4 -> 1 C2H2 1 CH2
1 C2H2 1 CH2(S) -> 1 H2CCCH 1 H
1 H2CCCH 1 H -> 1 C2H2 1 CH2(S)
1 C2H2 1 C2H -> 1 C4H2 1 H
1 C4H2 1 H -> 1 C2H2 1 C2H
1 C2H2 1 O -> 1 CH2 1 CO
1 CH2 1 CO -> 1 C2H2 1 O
1 C2H2 1 O -> 1 HCCO 1 H
1 HCCO 1 H -> 1 C2H2 1 O
1 C2H2 1 OH -> 1 C2H 1 H2O
1 C2H 1 H2O -> 1 C2H2 1 OH
1 C2H2 -> 1 C2H 1 H
1 C2H 1 H -> 1 C2H2
1 C2H4 1 H -> 1 C2H3 1 H2
1 C2H3 1 H2 -> 1 C2H4 1 H
1 C2H4 1 CH -> 1 C3H4 1 H
1 C3H4 1 H -> 1 C2H4 1 CH
1 C2H4 1 CH2(S) -> 1 C3H6
1 C3H6 -> 1 C2H4 1 CH2(S)
1 C2H4 1 CH3 -> 1 CH4 1 C2H3
1 CH4 1 C2H3 -> 1 C2H4 1 CH3
1 C2H4 1 O -> 1 H 1 CH2HCO
1 H 1 CH2HCO -> 1 C2H4 1 O
1 C2H4 1 O -> 1 CH3 1 HCO
1 CH3 1 HCO -> 1 C2H4 1 O
1 C2H4 1 O -> 1 CH2CO 1 H2
1 CH2CO 1 H2 -> 1 C2H4 1 O
1 C2H4 1 OH -> 1 C2H3 1 H2O
1 C2H3 1 H2O -> 1 C2H4 1 OH
1 C2H4 -> 1 C2H2 1 H2
1 C2H2 1 H2 -> 1 C2H4
1 C2H4 -> 1 C2H3 1 H
1 C2H3 1 H -> 1 C2H4
1 C2H6 1 H -> 1 C2H5 1 H2
1 C2H5 1 H2 -> 1 C2H6 1 H
1 C2H6 1 CH -> 1 C2H4 1 CH3
1 C2H4 1 CH3 -> 1 C2H6 1 CH
1 C2H6 1 CH2(S) -> 1 CH3 1 C2H5
1 CH3 1 C2H5 -> 1 C2H6 1 CH2(S)
1 C2H6 1 CH3 -> 1 C2H5 1 CH4
1 C2H5 1 CH4 -> 1 C2H6 1 CH3
```

```
1 C2H6 1 O -> 1 C2H5 1 OH
1 C2H5 1 OH -> 1 C2H6 1 O
1 C2H6 1 OH -> 1 C2H5 1 H2O
1 C2H5 1 H2O -> 1 C2H6 1 OH
1 C2H6 1 HO2 -> 1 H2O2 1 C2H5
1 H2O2 1 C2H5 -> 1 C2H6 1 HO2
1 C4H2 1 O -> 1 C3H2 1 CO
1 C3H2 1 CO -> 1 C4H2 1 O
1 C4H2 1 OH -> 1 C3H2 1 HCO
1 C3H2 1 HCO -> 1 C4H2 1 OH
1 O2 1 CO -> 1 CO2 1 O
1 CO2 1 O -> 1 O2 1 CO
1 O2 1 CH2O -> 1 HCO 1 HO2
1 HCO 1 HO2 -> 1 O2 1 CH2O
1 O2 1 C -> 1 CO 1 O
1 CO 1 O -> 1 O2 1 C
1 O2 1 H -> 1 HO2
1 HO2 -> 1 O2 1 H
1 O2 1 H 1 H2O -> 1 HO2 1 H2O
1 HO2 1 H2O -> 1 O2 1 H 1 H2O
1 O2 1 H -> 1 OH 1 O
1 OH 1 O -> 1 O2 1 H
1 O2 1 CH -> 1 CO 1 OH
1 CO 1 OH -> 1 O2 1 CH
1 O2 1 CH -> 1 CO2 1 H
1 CO2 1 H -> 1 O2 1 CH
1 O2 1 CH2 -> 1 CO2 1 H2
1 CO2 1 H2 -> 1 O2 1 CH2
1 O2 1 CH2 -> 1 CO2 1 H 1 H
1 CO2 1 H 1 H -> 1 O2 1 CH2
1 O2 1 CH2 -> 1 CO 1 OH 1 H
1 CO 1 OH 1 H -> 1 O2 1 CH2
1 O2 1 CH2 -> 1 CO 1 H2O
1 CO 1 H2O -> 1 O2 1 CH2
1 O2 1 CH2 -> 1 CH2O 1 O
1 CH2O 1 O -> 1 O2 1 CH2
1 O2 1 CH2(S) -> 1 CO 1 OH 1 H
1 CO 1 OH 1 H -> 1 O2 1 CH2(S)
1 O2 1 CH3 -> 1 CH2O 1 OH
1 CH2O 1 OH -> 1 O2 1 CH3
1 O2 1 C2H -> 1 HCCO 1 O
1 HCCO 1 O -> 1 O2 1 C2H
1 O2 1 C2H -> 1 CO2 1 CH
1 CO2 1 CH -> 1 O2 1 C2H
1 O2 1 C2H3 -> 1 C2H2 1 HO2
1 C2H2 1 HO2 -> 1 O2 1 C2H3
1 O2 1 C2H5 -> 1 C2H4 1 HO2
1 C2H4 1 HO2 -> 1 O2 1 C2H5
1 O2 1 C3H2 -> 1 HCO 1 HCCO
1 HCO 1 HCCO -> 1 O2 1 C3H2
1 O2 1 H2CCCH -> 1 CH2CO 1 HCO
1 CH2CO 1 HCO -> 1 O2 1 H2CCCH
1 O2 1 HCO -> 1 HO2 1 CO
1 HO2 1 CO -> 1 O2 1 HCO
1 O2 1 CH3O -> 1 CH2O 1 HO2
1 CH2O 1 HO2 -> 1 O2 1 CH3O
1 O2 1 CH2OH -> 1 CH2O 1 HO2
1 CH2O 1 HO2 -> 1 O2 1 CH2OH
1 O2 1 CH2OH -> 1 CH2O 1 HO2
1 CH2O 1 HO2 -> 1 O2 1 CH2OH
1 O2 1 HCCO -> 1 CO 1 CO 1 OH
1 CO 1 CO 1 OH -> 1 O2 1 HCCO
1 H2O2 1 H -> 1 HO2 1 H2
1 HO2 1 H2 -> 1 H2O2 1 H
1 H2O2 1 H -> 1 OH 1 H2O
1 OH 1 H2O -> 1 H2O2 1 H
1 H2O2 1 O -> 1 OH 1 HO2
1 OH 1 HO2 -> 1 H2O2 1 O
1 H2O2 1 OH -> 1 H2O 1 HO2
1 H2O 1 HO2 -> 1 H2O2 1 OH
1 CO 1 O -> 1 CO2
1 CO2 -> 1 CO 1 O
1 CO 1 OH -> 1 CO2 1 H
1 CO2 1 H -> 1 CO 1 OH
1 CO 1 HO2 -> 1 CO2 1 OH
1 CO2 1 OH -> 1 CO 1 HO2
1 CO 1 CH -> 1 HCCO
1 HCCO -> 1 CO 1 CH
1 CO2 1 CH -> 1 HCO 1 CO
1 HCO 1 CO -> 1 CO2 1 CH
1 CO2 1 CH2 -> 1 CH2O 1 CO
1 CH2O 1 CO -> 1 CO2 1 CH2
1 CH2O 1 H -> 1 HCO 1 H2
1 HCO 1 H2 -> 1 CH2O 1 H
1 CH2O 1 CH -> 1 CH2 1 HCO
1 CH2 1 HCO -> 1 CH2O 1 CH
1 CH2O 1 CH3 -> 1 CH4 1 HCO
1 CH4 1 HCO -> 1 CH2O 1 CH3
1 CH2O 1 O -> 1 HCO 1 OH
1 HCO 1 OH -> 1 CH2O 1 O
1 CH2O 1 OH -> 1 HCO 1 H2O
1 HCO 1 H2O -> 1 CH2O 1 OH
1 CH2O 1 HO2 -> 1 H2O2 1 HCO
1 H2O2 1 HCO -> 1 CH2O 1 HO2
1 CH2O -> 1 HCO 1 H
1 HCO 1 H -> 1 CH2O
1 CH2O -> 1 H2 1 CO
1 H2 1 CO -> 1 CH2O
1 CH2CO 1 H -> 1 CH3 1 CO
1 CH3 1 CO -> 1 CH2CO 1 H
1 CH2CO 1 O -> 1 CH2 1 CO2
1 CH2 1 CO2 -> 1 CH2CO 1 O
1 CH2CO 1 O -> 1 CH2O 1 CO
1 CH2O 1 CO -> 1 CH2CO 1 O
1 CH2CO 1 O -> 1 HCO 1 H 1 CO
1 HCO 1 H 1 CO -> 1 CH2CO 1 O
1 CH2CO 1 O -> 1 HCO 1 HCO
1 HCO 1 HCO -> 1 CH2CO 1 O
1 CH2CO 1 OH -> 1 CH3 1 CO2
1 CH3 1 CO2 -> 1 CH2CO 1 OH
1 CH2CO 1 OH -> 1 CH2OH 1 CO
1 CH2OH 1 CO -> 1 CH2CO 1 OH
1 CH2CO -> 1 CH2 1 CO
1 CH2 1 CO -> 1 CH2CO
1 CH2CO -> 1 HCCO 1 H
1 HCCO 1 H -> 1 CH2CO
1 C 1 CH2 -> 1 C2H 1 H
1 C2H 1 H -> 1 C 1 CH2
1 C 1 CH3 -> 1 C2H2 1 H
1 C2H2 1 H -> 1 C 1 CH3
1 C 1 OH -> 1 CO 1 H
1 CO 1 H -> 1 C 1 OH
1 H 1 H -> 1 H2
1 H2 -> 1 H 1 H
1 H 1 H 1 H2 -> 1 H2 1 H2
1 H2 1 H2 -> 1 H 1 H 1 H2
1 H 1 CH -> 1 C 1 H2
1 C 1 H2 -> 1 H 1 CH
1 H 1 CH2 -> 1 CH 1 H2
1 CH 1 H2 -> 1 H 1 CH2
1 H 1 CH2(S) -> 1 CH2 1 H
1 CH2 1 H -> 1 H 1 CH2(S)
1 H 1 C2H3 -> 1 C2H2 1 H2
1 C2H2 1 H2 -> 1 H 1 C2H3
1 CH3 1 CH3 -> 1 C2H5 1 H
1 C2H5 1 H -> 1 CH3 1 CH3
1 H 1 O -> 1 OH
1 OH -> 1 H 1 O
1 H 1 OH -> 1 H2O
1 H2O -> 1 H 1 OH
1 H 1 HO2 -> 1 H2 1 O2
1 H2 1 O2 -> 1 H 1 HO2
1 H 1 HO2 -> 1 OH 1 OH
1 OH 1 OH -> 1 H 1 HO2
1 H 1 HO2 -> 1 H2O 1 O
1 H2O 1 O -> 1 H 1 HO2
```

```
1 H 1 HCO -> 1 CO 1 H2
1 CO 1 H2 -> 1 H 1 HCO
1 H 1 CH3O -> 1 CH2O 1 H2
1 CH2O 1 H2 -> 1 H 1 CH3O
1 H 1 CH2OH -> 1 CH3 1 OH
1 CH3 1 OH -> 1 H 1 CH2OH
1 H 1 CH2OH -> 1 CH2O 1 H2
1 CH2O 1 H2 -> 1 H 1 CH2OH
1 H 1 HCCO -> 1 CH2 1 CO
1 CH2 1 CO -> 1 H 1 HCCO
1 CH 1 CH2 -> 1 C2H2 1 H
1 C2H2 1 H -> 1 CH 1 CH2
1 CH 1 CH3 -> 1 C2H3 1 H
1 C2H3 1 H -> 1 CH 1 CH3
1 CH 1 C2H3 -> 1 CH2 1 C2H2
1 CH2 1 C2H2 -> 1 CH 1 C2H3
1 CH 1 O -> 1 CO 1 H
1 CO 1 H -> 1 CH 1 O
1 CH 1 OH -> 1 HCO 1 H
1 HCO 1 H -> 1 CH 1 OH
1 CH 1 HCCO -> 1 C2H2 1 CO
1 C2H2 1 CO -> 1 CH 1 HCCO
1 CH2 1 CH2 -> 1 C2H2 1 H2
1 C2H2 1 H2 -> 1 CH2 1 CH2
1 CH2 1 CH2 -> 1 C2H2 1 H 1 H
1 C2H2 1 H 1 H -> 1 CH2 1 CH2
1 CH2 1 CH3 -> 1 C2H4 1 H
1 C2H4 1 H -> 1 CH2 1 CH3
1 CH2 1 C2H3 -> 1 C2H2 1 CH3
1 C2H2 1 CH3 -> 1 CH2 1 C2H3
1 CH2 1 O -> 1 CO 1 H 1 H
1 CO 1 H 1 H -> 1 CH2 1 O
1 CH2 1 O -> 1 CO 1 H2
1 CO 1 H2 -> 1 CH2 1 O
1 CH2 1 OH -> 1 CH2O 1 H
1 CH2O 1 H -> 1 CH2 1 OH
1 CH2 1 HCO -> 1 CH3 1 CO
1 CH3 1 CO -> 1 CH2 1 HCO
1 CH2 1 HCCO -> 1 C2H3 1 CO
1 C2H3 1 CO -> 1 CH2 1 HCCO
1 CH2 1 HCCO -> 1 C2H 1 CH2O
1 C2H 1 CH2O -> 1 CH2 1 HCCO
1 CH2(S) -> 1 CH2
1 CH2 -> 1 CH2(S)
1 CH3 1 O -> 1 CH2O 1 H
1 CH2O 1 H -> 1 CH3 1 O
1 CH3 1 OH -> 1 CH2(S) 1 H2O
1 CH2(S) 1 H2O -> 1 CH3 1 OH
1 CH3 1 HO2 -> 1 CH3O 1 OH
1 CH3O 1 OH -> 1 CH3 1 HO2
1 CH3 1 HCO -> 1 CH4 1 CO
1 CH4 1 CO -> 1 CH3 1 HCO
1 CH3 -> 1 CH2 1 H
1 CH2 1 H -> 1 CH3
1 C2H 1 C2H3 -> 1 C2H2 1 C2H2
1 C2H2 1 C2H2 -> 1 C2H 1 C2H3
1 C2H 1 O -> 1 CH 1 CO
1 CH 1 CO -> 1 C2H 1 O
1 C2H 1 OH -> 1 HCCO 1 H
1 HCCO 1 H -> 1 C2H 1 OH
1 C2H 1 OH -> 1 CH2 1 CO
1 CH2 1 CO -> 1 C2H 1 OH
1 C2H3 1 O -> 1 CO 1 CH3
1 CO 1 CH3 -> 1 C2H3 1 O
1 C2H3 1 OH -> 1 C2H2 1 H2O
1 C2H2 1 H2O -> 1 C2H3 1 OH
1 C2H5 1 O -> 1 CH2O 1 CH3
1 CH2O 1 CH3 -> 1 C2H5 1 O
1 H2CCCH 1 O -> 1 C2H2 1 CO 1 H
1 C2H2 1 CO 1 H -> 1 H2CCCH 1 O
1 H2CCCH 1 OH -> 1 C3H2 1 H2O
1 C3H2 1 H2O -> 1 H2CCCH 1 OH
1 H2CCCCH -> 1 C4H2 1 H
1 C4H2 1 H -> 1 H2CCCCH
1 O 1 O -> 1 O2
1 O2 -> 1 O 1 O
1 O 1 HO2 -> 1 O2 1 OH
1 O2 1 OH -> 1 O 1 HO2
1 O 1 HCO -> 1 CO 1 OH
1 CO 1 OH -> 1 O 1 HCO
1 O 1 HCO -> 1 CO2 1 H
1 CO2 1 H -> 1 O 1 HCO
1 O2 1 CH3 -> 1 CH3O 1 O
1 CH3O 1 O -> 1 O2 1 CH3
1 O 1 CH3O -> 1 CH2O 1 OH
1 CH2O 1 OH -> 1 O 1 CH3O
1 O 1 CH2OH -> 1 CH2O 1 OH
1 CH2O 1 OH -> 1 O 1 CH2OH
1 O 1 HCCO -> 1 H 1 CO 1 CO
1 H 1 CO 1 CO -> 1 O 1 HCCO
1 OH 1 OH -> 1 O 1 H2O
1 O 1 H2O -> 1 OH 1 OH
1 OH 1 HO2 -> 1 H2O 1 O2
1 H2O 1 O2 -> 1 OH 1 HO2
1 OH 1 HCO -> 1 H2O 1 CO
1 H2O 1 CO -> 1 OH 1 HCO
1 OH 1 CH3O -> 1 CH2O 1 H2O
1 CH2O 1 H2O -> 1 OH 1 CH3O
1 OH 1 CH2OH -> 1 CH2O 1 H2O
1 CH2O 1 H2O -> 1 OH 1 CH2OH
1 OH 1 HCCO -> 1 HCO 1 HCO
1 HCO 1 HCO -> 1 OH 1 HCCO
1 OH 1 HCCO -> 1 CH2O 1 CO
1 CH2O 1 CO -> 1 OH 1 HCCO
1 HO2 1 HO2 -> 1 H2O2 1 O2
1 H2O2 1 O2 -> 1 HO2 1 HO2
1 HO2 1 HO2 -> 1 H2O2 1 O2
1 H2O2 1 O2 -> 1 HO2 1 HO2
1 HCO 1 HCO -> 1 CH2O 1 CO
1 CH2O 1 CO -> 1 HCO 1 HCO
1 HCO -> 1 H 1 CO
1 H 1 CO -> 1 HCO
1 CH3O -> 1 CH2O 1 H
1 CH2O 1 H -> 1 CH3O
1 CH2OH -> 1 CH2O 1 H
1 CH2O 1 H -> 1 CH2OH
1 HCCO 1 HCCO -> 1 C2H2 1 CO 1 CO
1 C2H2 1 CO 1 CO -> 1 HCCO 1 HCCO
```

E 3 Artificial chemistry NTOP

```
# reactions rulesFrom: ntop.tab
# (generated by tab2rules)
# number of molecules:
16
# molecules:
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

```
# number of rules:
207
# rules:
1 1 1 1 -> 1 1
1 1 1 3 -> 1 1
1 1 1 4 -> 1 4
1 1 1 5 -> 1 5
1 1 1 6 -> 1 4
1 1 1 7 -> 1 5
1 1 1 9 -> 1 1
1 1 1 11 -> 1 1
1 1 1 12 -> 1 4
1 1 1 13 -> 1 5
1 1 1 14 -> 1 4
1 1 1 15 -> 1 5
1 2 1 2 -> 1 1
1 2 1 3 -> 1 1
1 2 1 6 -> 1 1
1 2 1 7 -> 1 1
1 2 1 8 -> 1 4
1 2 1 9 -> 1 4
1 2 1 10 -> 1 5
1 2 1 11 -> 1 5
1 2 1 12 -> 1 4
1 2 1 13 -> 1 4
1 2 1 14 -> 1 5
1 2 1 15 -> 1 5
1 3 1 1 -> 1 1
1 3 1 2 -> 1 1
1 3 1 3 -> 1 1
1 3 1 4 -> 1 4
1 3 1 5 -> 1 5
1 3 1 6 -> 1 5
1 3 1 7 -> 1 5
1 3 1 8 -> 1 4
1 3 1 9 -> 1 5
1 3 1 10 -> 1 5
1 3 1 11 -> 1 5
1 3 1 12 -> 1 4
1 3 1 13 -> 1 5
1 3 1 14 -> 1 5
1 3 1 15 -> 1 5
1 4 1 1 -> 1 2
1 4 1 3 -> 1 2
1 4 1 4 -> 1 8
1 4 1 5 -> 1 10
1 4 1 6 -> 1 8
1 4 1 7 -> 1 10
1 4 1 9 -> 1 2
1 4 1 11 -> 1 2
1 4 1 12 -> 1 8
1 4 1 13 -> 1 10
1 4 1 14 -> 1 8
1 4 1 15 -> 1 10
1 5 1 9 -> 1 3
1 5 1 1 -> 1 3
1 5 1 3 -> 1 3
1 5 1 4 -> 1 12
1 5 1 5 -> 1 15
1 5 1 6 -> 1 12
1 5 1 7 -> 1 15
1 5 1 11 -> 1 3
1 5 1 12 -> 1 12
1 5 1 13 -> 1 15
1 5 1 14 -> 1 12
1 5 1 15 -> 1 15
1 6 1 1 -> 1 2
1 6 1 2 -> 1 1
1 6 1 3 -> 1 3
1 6 1 4 -> 1 8
1 6 1 5 -> 1 10
1 6 1 6 -> 1 9
1 6 1 7 -> 1 11
1 6 1 8 -> 1 4
1 6 1 9 -> 1 6
1 6 1 10 -> 1 5
1 6 1 11 -> 1 7
1 6 1 12 -> 1 12
1 6 1 13 -> 1 14
1 6 1 14 -> 1 13
1 6 1 15 -> 1 15
1 7 1 1 -> 1 3
1 7 1 2 -> 1 1
1 7 1 3 -> 1 3
1 7 1 4 -> 1 12
1 7 1 5 -> 1 15
1 7 1 6 -> 1 13
1 7 1 7 -> 1 15
1 7 1 8 -> 1 4
1 7 1 9 -> 1 7
1 7 1 10 -> 1 5
1 7 1 11 -> 1 7
1 7 1 12 -> 1 12
1 7 1 13 -> 1 15
1 7 1 14 -> 1 13
1 7 1 15 -> 1 15
1 8 1 2 -> 1 2
1 8 1 3 -> 1 2
1 8 1 6 -> 1 2
1 8 1 7 -> 1 2
1 8 1 8 -> 1 8
1 8 1 9 -> 1 8
1 8 1 10 -> 1 10
1 8 1 11 -> 1 10
1 8 1 12 -> 1 8
1 8 1 13 -> 1 8
1 8 1 14 -> 1 10
1 8 1 15 -> 1 10
1 9 1 1 -> 1 1
1 9 1 2 -> 1 2
1 9 1 3 -> 1 3
1 9 1 4 -> 1 4
1 9 1 5 -> 1 5
1 9 1 6 -> 1 6
1 9 1 7 -> 1 7
1 9 1 8 -> 1 8
1 9 1 9 -> 1 9
1 9 1 10 -> 1 10
1 9 1 11 -> 1 11
1 9 1 12 -> 1 12
1 9 1 13 -> 1 13
1 9 1 14 -> 1 14
1 9 1 15 -> 1 15
1 10 1 2 -> 1 3
1 10 1 3 -> 1 3
1 10 1 6 -> 1 3
1 10 1 7 -> 1 3
1 10 1 8 -> 1 12
1 10 1 9 -> 1 12
1 10 1 10 -> 1 15
1 10 1 11 -> 1 15
1 10 1 12 -> 1 12
1 10 1 13 -> 1 12
1 10 1 14 -> 1 15
1 10 1 15 -> 1 15
1 11 1 1 -> 1 1
1 11 1 2 -> 1 3
1 11 1 3 -> 1 3
1 11 1 4 -> 1 4
1 11 1 5 -> 1 5
1 11 1 6 -> 1 7
1 11 1 7 -> 1 7
1 11 1 8 -> 1 12
1 11 1 9 -> 1 13
1 11 1 10 -> 1 15
1 11 1 11 -> 1 15
```

```
1 11 1 12 -> 1 12
1 11 1 13 -> 1 13
1 11 1 14 -> 1 15
1 11 1 15 -> 1 15
1 12 1 1 -> 1 2
1 12 1 2 -> 1 2
1 12 1 3 -> 1 2
1 12 1 4 -> 1 8
1 12 1 5 -> 1 10
1 12 1 6 -> 1 10
1 12 1 7 -> 1 10
1 12 1 8 -> 1 8
1 12 1 9 -> 1 10
1 12 1 10 -> 1 10
1 12 1 11 -> 1 10
1 12 1 12 -> 1 8
1 12 1 13 -> 1 10
1 12 1 14 -> 1 10
1 12 1 15 -> 1 10
1 13 1 1 -> 1 3
1 13 1 2 -> 1 2
1 13 1 3 -> 1 3
1 13 1 4 -> 1 12
1 13 1 5 -> 1 15
1 13 1 6 -> 1 14
1 13 1 7 -> 1 15
1 13 1 8 -> 1 8
1 13 1 9 -> 1 11
1 13 1 10 -> 1 10
1 13 1 11 -> 1 11
1 13 1 12 -> 1 12
1 13 1 13 -> 1 15
1 13 1 14 -> 1 14
1 13 1 15 -> 1 15
1 14 1 1 -> 1 2
1 14 1 2 -> 1 3
1 14 1 3 -> 1 3
1 14 1 4 -> 1 8
1 14 1 5 -> 1 10
1 14 1 6 -> 1 11
1 14 1 7 -> 1 11
1 14 1 8 -> 1 12
1 14 1 9 -> 1 14
1 14 1 10 -> 1 15
1 14 1 11 -> 1 15
1 14 1 12 -> 1 12
1 14 1 13 -> 1 14
1 14 1 14 -> 1 15
1 14 1 15 -> 1 15
1 15 1 1 -> 1 3
1 15 1 2 -> 1 3
1 15 1 3 -> 1 3
1 15 1 4 -> 1 12
1 15 1 5 -> 1 15
1 15 1 6 -> 1 15
1 15 1 7 -> 1 15
1 15 1 8 -> 1 12
1 15 1 9 -> 1 15
1 15 1 10 -> 1 15
1 15 1 11 -> 1 15
1 15 1 12 -> 1 12
1 15 1 13 -> 1 15
1 15 1 14 -> 1 15
1 15 1 15 -> 1 15
```

E 4 Gene translation

E 4.1 NCBI Merge

```
# number of molecules:
234
# molecules:
A
AAA
AAC
AAG
AAT
ACA
ACC
ACG
ACT
AGA
AGC
AGG
AGT
ATA
ATC
ATG
ATT
C
CAA
CAC
CAG
CAT
CCA
CCC
CCG
CCT
CGA
CGC
CGG
CGT
CTA
CTC
CTG
CTT
D
E
F
G
GAA
GAC
GAG
GAT
GCA
GCC
GCG
GCT
GGA
GGC
GGG
GGT
GTA
```

```
GTC
GTG
GTT
H
I
K
L
M
N
O
P
Q
R
S
T
TAA
TAC
TAG
TAT
TCA
TCC
TCG
TCT
TGA
TGC
TGG
TGT
TTA
TTC
TTG
TTT
V
W
Y
tRNAAAA
tRNAAAAK
tRNAAAAN
tRNAAAC
tRNAAACN
tRNAAAG
tRNAAAGK
tRNAAAT
tRNAAATN
tRNAACA
tRNAACAT
tRNAACC
tRNAACCT
tRNAACG
tRNAACGT
tRNAACT
tRNAACTT
tRNAAGA
tRNAAGAG
tRNAAGAO
tRNAAGAR
tRNAAGAS
tRNAAGC
tRNAAGCS
tRNAAGG
tRNAAGGG
tRNAAGGO
tRNAAGGR
tRNAAGGS
tRNAAGT
tRNAAGTS
tRNAATA
tRNAATAI
tRNAATAM
tRNAATC
tRNAATCI
tRNAATG
tRNAATGM
tRNAATT
tRNAATTI
tRNACAA
tRNACAAQ
tRNACAC
tRNACACH
tRNACAG
tRNACAGQ
tRNACAT
tRNACATH
tRNACCA
tRNACCAP
tRNACCC
tRNACCCP
tRNACCG
tRNACCGP
tRNACCT
tRNACCTP
tRNACGA
tRNACGAR
tRNACGC
tRNACGCR
tRNACGG
tRNACGGR
tRNACGT
tRNACGTR
tRNACTA
tRNACTAL
tRNACTAT
tRNACTC
tRNACTCL
tRNACTCT
tRNACTG
tRNACTGL
tRNACTGS
tRNACTGT
tRNACTT
tRNACTTL
tRNACTTT
tRNAGAA
tRNAGAAE
tRNAGAC
tRNAGACD
tRNAGAG
```

```
tRNAGAGE
tRNAGAT
tRNAGATD
tRNAGCA
tRNAGCAA
tRNAGCC
tRNAGCCA
tRNAGCG
tRNAGCGA
tRNAGCT
tRNAGCTA
tRNAGGA
tRNAGGAG
tRNAGGC
tRNAGGCG
tRNAGGG
tRNAGGGG
tRNAGGT
tRNAGGTG
tRNAGTA
tRNAGTAV
tRNAGTC
tRNAGTCV
tRNAGTG
tRNAGTGV
tRNAGTT
tRNAGTTV
tRNATAA
tRNATAAO
tRNATAAQ
tRNATAAY
tRNATAC
tRNATACY
tRNATAG
tRNATAGL
tRNATAGO
tRNATAGQ
tRNATAT
tRNATATY
tRNATCA
tRNATCAO
tRNATCAS
tRNATCC
tRNATCCS
tRNATCG
tRNATCGS
tRNATCT
tRNATCTS
tRNATGA
tRNATGAC
tRNATGAO
tRNATGAW
tRNATGC
tRNATGCC
tRNATGG
tRNATGGW
tRNATGT
tRNATGTC
tRNATTA
tRNATTAL
tRNATTAO
tRNATTC
tRNATTCF
tRNATTG
tRNATTGL
tRNATTT
tRNATTTF
# number of rules:
85
# rules:
1 CTG 1 tRNACTGT -> 1 T
1 GAC 1 tRNAGACD -> 1 D
1 TAG 1 tRNATAGO -> 1 O
1 TAC 1 tRNATACY -> 1 Y
1 CTC 1 tRNACTCL -> 1 L
1 GAG 1 tRNAGAGE -> 1 E
1 GTA 1 tRNAGTAV -> 1 V
1 AGG 1 tRNAAGGG -> 1 G
1 AGA 1 tRNAAGAO -> 1 O
1 TCC 1 tRNATCCS -> 1 S
1 AGT 1 tRNAAGTS -> 1 S
1 TAG 1 tRNATAGQ -> 1 Q
1 ACA 1 tRNAACAT -> 1 T
1 GCG 1 tRNAGCGA -> 1 A
1 CTC 1 tRNACTCT -> 1 T
1 CCC 1 tRNACCCP -> 1 P
1 TAA 1 tRNATAAO -> 1 O
1 CTT 1 tRNACTTL -> 1 L
1 CTG 1 tRNACTGS -> 1 S
1 TTT 1 tRNATTTF -> 1 F
1 GGT 1 tRNAGGTG -> 1 G
1 GAT 1 tRNAGATD -> 1 D
1 CGG 1 tRNACGGR -> 1 R
1 ATT 1 tRNAATTI -> 1 I
1 CTG 1 tRNACTGL -> 1 L
1 ATA 1 tRNAATAI -> 1 I
1 ACT 1 tRNAACTT -> 1 T
1 GTT 1 tRNAGTTV -> 1 V
1 GCT 1 tRNAGCTA -> 1 A
1 GCA 1 tRNAGCAA -> 1 A
1 TAA 1 tRNATAAY -> 1 Y
1 CAT 1 tRNACATH -> 1 H
1 ATA 1 tRNAATAM -> 1 M
1 TCG 1 tRNATCGS -> 1 S
1 ATG 1 tRNAATGM -> 1 M
1 TGA 1 tRNATGAW -> 1 W
1 GAA 1 tRNAGAAE -> 1 E
1 AAA 1 tRNAAAAN -> 1 N
1 TCA 1 tRNATCAS -> 1 S
1 AAA 1 tRNAAAAK -> 1 K
1 TCA 1 tRNATCAO -> 1 O
1 TAT 1 tRNATATY -> 1 Y
1 TGA 1 tRNATGAC -> 1 C
1 AGA 1 tRNAAGAR -> 1 R
1 CTA 1 tRNACTAL -> 1 L
1 AGA 1 tRNAAGAS -> 1 S
```

```
1 TGT 1 tRNATGTC -> 1 C
1 CTA 1 tRNACTAT -> 1 T
1 TTC 1 tRNATTCF -> 1 F
1 CCT 1 tRNACCTP -> 1 P
1 CGT 1 tRNACGTR -> 1 R
1 CGA 1 tRNACGAR -> 1 R
1 TGC 1 tRNATGCC -> 1 C
1 CCA 1 tRNACCAP -> 1 P
1 AAG 1 tRNAAAGK -> 1 K
1 GCC 1 tRNAGCCA -> 1 A
1 CAG 1 tRNACAGQ -> 1 Q
1 TGA 1 tRNATGAO -> 1 O
1 GTC 1 tRNAGTCV -> 1 V
1 AGA 1 tRNAAGAG -> 1 G
1 TTG 1 tRNATTGL -> 1 L
1 TCT 1 tRNATCTS -> 1 S
1 ACG 1 tRNAACGT -> 1 T
1 TGG 1 tRNATGGW -> 1 W
1 AAC 1 tRNAAACN -> 1 N
1 GGG 1 tRNAGGGG -> 1 G
1 CAA 1 tRNACAAQ -> 1 Q
1 TAA 1 tRNATAAQ -> 1 Q
1 AGG 1 tRNAAGGR -> 1 R
1 TTA 1 tRNATTAO -> 1 O
1 AGG 1 tRNAAGGS -> 1 S
1 TAG 1 tRNATAGL -> 1 L
1 ACC 1 tRNAACCT -> 1 T
1 GGC 1 tRNAGGCG -> 1 G
1 AAT 1 tRNAAATN -> 1 N
1 GGA 1 tRNAGGAG -> 1 G
1 CTT 1 tRNACTTT -> 1 T
1 CCG 1 tRNACCGP -> 1 P
1 CGC 1 tRNACGCR -> 1 R
1 AGC 1 tRNAAGCS -> 1 S
1 CAC 1 tRNACACH -> 1 H
1 GTG 1 tRNAGTGV -> 1 V
1 TTA 1 tRNATTAL -> 1 L
1 ATC 1 tRNAATCI -> 1 I
1 AGG 1 tRNAAGGO -> 1 O
```

E 4.2 Completed GC w/o synthetases (excerpt)

```
# number of molecules:
1364
# molecules:
C1
C2
...
C63
C64
AAprot1
AAprot2
...
AAprot19
AAprot20
tRNA11
tRNA12
...
tRNA641
tRNA642
tRNA643
tRNA644
tRNA645
tRNA646
tRNA647
tRNA648
tRNA649
tRNA6410
tRNA6411
tRNA6412
tRNA6413
tRNA6414
tRNA6415
tRNA6416
tRNA6417
tRNA6418
tRNA6419
tRNA6420
# number of rules:
1280
# rules:
1 tRNA11 1 C1 -> 1 C1 1 AAprot1
1 tRNA12 1 C1 -> 1 C1 1 AAprot2
...
1 tRNA6410 1 C64 -> 1 C64 1 AAprot10
1 tRNA6411 1 C64 -> 1 C64 1 AAprot11
1 tRNA6412 1 C64 -> 1 C64 1 AAprot12
1 tRNA6413 1 C64 -> 1 C64 1 AAprot13
1 tRNA6414 1 C64 -> 1 C64 1 AAprot14
1 tRNA6415 1 C64 -> 1 C64 1 AAprot15
1 tRNA6416 1 C64 -> 1 C64 1 AAprot16
1 tRNA6417 1 C64 -> 1 C64 1 AAprot17
1 tRNA6418 1 C64 -> 1 C64 1 AAprot18
1 tRNA6419 1 C64 -> 1 C64 1 AAprot19
1 tRNA6420 1 C64 -> 1 C64 1 AAprot20
```

E 4.3 Complete GC with synthetases (excerpt)

```
# number of molecules:
2728
# molecules:
C1
tRNA1
C2
tRNA2
...
C63
tRNA63
C64
tRNA64
AA1-free
AA1-prot
AA2-free
AA2-prot
...
AA19-free
AA19-prot
AA20-free
AA20-prot
Syn_C1-AA1
AA1-tRNA1
Syn_C1-AA2
AA2-tRNA1
...
Syn_C64-AA19
AA19-tRNA64
Syn_C64-AA20
AA20-tRNA64
# number of rules:
2560
```

```
# rules:
1 AA1-free 1 tRNA1 1 Syn_C1-AA1
                -> 1 Syn_C1-AA1 1 AA1-tRNA1
1 AA1-tRNA1 1 C1 -> 1 C1 1 AA1-prot 1 tRNA1
1 AA2-free 1 tRNA1 1 Syn_C1-AA2
                -> 1 Syn_C1-AA2 1 AA2-tRNA1
1 AA2-tRNA1 1 C1 -> 1 C1 1 AA2-prot 1 tRNA1
...
1 AA17-free 1 tRNA64 1 Syn_C64-AA17
                  -> 1 Syn_C64-AA17 1 AA17-tRNA64
1 AA17-tRNA64 1 C64 -> 1 C64 1 AA17-prot 1 tRNA64
1 AA18-free 1 tRNA64 1 Syn_C64-AA18
                  -> 1 Syn_C64-AA18 1 AA18-tRNA64
1 AA18-tRNA64 1 C64 -> 1 C64 1 AA18-prot 1 tRNA64
1 AA19-free 1 tRNA64 1 Syn_C64-AA19
                  -> 1 Syn_C64-AA19 1 AA19-tRNA64
1 AA19-tRNA64 1 C64 -> 1 C64 1 AA19-prot 1 tRNA64
1 AA20-free 1 tRNA64 1 Syn_C64-AA20
                  -> 1 Syn_C64-AA20 1 AA20-tRNA64
1 AA20-tRNA64 1 C64 -> 1 C64 1 AA20-prot 1 tRNA64
```

E 5 Gene regulatory networks

E 5.1 GC-GRN network

```
# Number of Components:
14
# Components:
TF1
TF2
tRNAAL
tRNAAK
tRNABL
tRNABK
P1ABA
P2ABA
P1BAB
P2BAB
LKL
KLK
LLL
KKK
# Number of Reactions:
16
# Reactions:
1 TF1 1 P1ABA 1 tRNAAL 1 tRNABK -> 1 LKL
1 TF1 1 P1ABA 1 tRNAAL 1 tRNABL -> 1 LLL
1 TF1 1 P1ABA 1 tRNAAK 1 tRNABK -> 1 KKK
1 TF1 1 P1ABA 1 tRNAAK 1 tRNABL -> 1 KLK
1 TF1 1 P1BAB 1 tRNAAL 1 tRNABK -> 1 KLK
1 TF1 1 P1BAB 1 tRNAAL 1 tRNABL -> 1 LLL
1 TF1 1 P1BAB 1 tRNAAK 1 tRNABK -> 1 KKK
1 TF1 1 P1BAB 1 tRNAAK 1 tRNABL -> 1 LKL
1 TF2 1 P2ABA 1 tRNAAL 1 tRNABK -> 1 LKL
1 TF2 1 P2ABA 1 tRNAAL 1 tRNABL -> 1 LLL
1 TF2 1 P2ABA 1 tRNAAK 1 tRNABK -> 1 KKK
1 TF2 1 P2ABA 1 tRNAAK 1 tRNABL -> 1 KLK
1 TF2 1 P2BAB 1 tRNAAL 1 tRNABK -> 1 KLK
1 TF2 1 P2BAB 1 tRNAAL 1 tRNABL -> 1 LLL
1 TF2 1 P2BAB 1 tRNAAK 1 tRNABK -> 1 KKK
1 TF2 1 P2BAB 1 tRNAAK 1 tRNABL -> 1 LKL
```

E 5.2 Extended GC-GRN network

```
# reactions extended GC GRN model
# number of molecules:
16
# molecules:
TF1
TF2
tRNAAL
tRNAAK
tRNABL
tRNABK
P1ABA
P2ABA
P1BAB
P2BAB
LKL
KLK
LLL
KKK
ABA
BAB
# number of rules:
10
# rules:
1 TF1 1 P1ABA -> 1 ABA
1 TF2 1 P2ABA -> 1 ABA
1 TF1 1 P1BAB -> 1 BAB
1 TF2 1 P2BAB -> 1 BAB
1 ABA 1 tRNAAL 1 tRNABK -> 1 LKL
1 ABA 1 tRNAAL 1 tRNABL -> 1 LLL
1 ABA 1 tRNAAK 1 tRNABK -> 1 KKK
1 BAB 1 tRNAAK 1 tRNABL -> 1 LKL
1 BAB 1 tRNAAL 1 tRNABK -> 1 KLK
1 BAB 1 tRNAAL 1 tRNABL -> 1 LLL
```

E 6 Phosphorylation cascades

E 6.1 Simple phosphorylation model

```
# Number of Components:
3
# Components:
A
AP
SP
# Number of Reactions:
2
```

```
# Reactions:
1 A 1 SP -> 1 AP 1 SP
1 AP -> 1 A
```

E 6.2 Extended phosphorylation model

```
# Number of Components:
7
# Components:
A
AP
B
BP
C
CP
SP
# Number of Reactions:
6
# Reactions:
1 B 1 SP -> 1 BP 1 SP
1 BP -> 1 B
1 C 1 SP -> 1 CP 1 SP
1 CP -> 1 C
1 A 1 B -> 1 AP
1 AP 1 CP -> 1 A
```

E 7 Protein assembly

E 7.1 Two steps, without dissociation

```
# Number of Components
20
# Components
A
B
AA
AB
BB
AAA
AAB
ABA
ABB
BAB
BBB
AAAA
AAAB
AABB
ABAA
ABAB
ABBA
ABBB
BBAB
BBBB
# Number of Reactions
20
# Reactions
1 A 1 A -> 1 AA
1 A 1 B -> 1 AB
1 B 1 B -> 1 BB
1 A 1 AA -> 1 AAA
1 A 1 AB -> 1 AAB
1 A 1 AB -> 1 ABA
1 A 1 BB -> 1 ABB
1 B 1 AA -> 1 AAB
1 B 1 AB -> 1 ABB
1 B 1 AB -> 1 BAB
1 B 1 BB -> 1 BBB
1 AA 1 AA -> 1 AAAA
1 AA 1 AB -> 1 AAAB
1 AA 1 AB -> 1 ABAA
1 AA 1 BB -> 1 AABB
1 AB 1 AB -> 1 ABAB
1 AB 1 AB -> 1 ABBA
1 BB 1 AB -> 1 ABBB
1 BB 1 AB -> 1 BBAB
1 BB 1 BB -> 1 BBBB
```

E 7.2 Two steps, with dissociation

```
# Number of Components
20
# Components
A
B
AA
AB
BB
AAA
AAB
ABA
ABB
BAB
BBB
AAAA
AAAB
AABB
ABAA
ABAB
ABBA
ABBB
BBAB
BBBB
# Number of Reactions
23
# Reactions
1 A 1 A -> 1 AA
1 AA -> 1 A 1 A
1 A 1 B -> 1 AB
1 AB -> 1 A 1 B
```

```
1 B 1 B -> 1 BB
1 BB -> 1 B 1 B
1 A 1 AA -> 1 AAA
1 A 1 AB -> 1 AAB
1 A 1 AB -> 1 ABA
1 A 1 BB -> 1 ABB
1 B 1 AA -> 1 AAB
1 B 1 AB -> 1 ABB
1 B 1 AB -> 1 BAB
1 B 1 BB -> 1 BBB
1 AA 1 AA -> 1 AAAA
1 AA 1 AB -> 1 AAAB
1 AA 1 AB -> 1 ABAA
1 AA 1 BB -> 1 AABB
1 AB 1 AB -> 1 ABAB
1 AB 1 AB -> 1 ABBA
1 BB 1 AB -> 1 ABBB
1 BB 1 AB -> 1 BBAB
1 BB 1 BB -> 1 BBBB
```

E 8 Photochemistry of Mars

```
# Number of Components
32
# Components
hv
M
e
O_3
O_2
O
O(^1D)
H_2
H
OH
HO_2
H_2O
H_2O_2
CO_2
CO
N_2
N
N(^2D)
NO
NO_2
NO_3
N_2O
N_2O_5
HNO_2
HNO_3
HO_2NO_2
O^+
O_2^+
CO_2^+
CO_2H^+
(HO_2)_grain
grain
# Number of Reactions
104
# Reactions
-> 1 hv
1 O_2 1 hv -> 2 O
1 O_2 1 hv -> 1 O 1 O(^1D)
1 O_3 1 hv -> 1 O_2 1 O
1 O_3 1 hv -> 1 O_2 1 O(^1D)
1 O_3 1 hv -> 3 O
1 H_2 1 hv -> 2 H
1 OH 1 hv -> 1 O 1 H
1 HO_2 1 hv -> 1 OH 1 O
1 H_2O 1 hv -> 1 H 1 OH
1 H_2O 1 hv -> 1 H_2 1 O(^1D)
1 H_2O 1 hv -> 2 H 1 O
1 H_2O_2 1 hv -> 2 OH
1 CO_2 1 hv -> 1 CO 1 O
1 CO_2 1 hv -> 1 CO 1 O(^1D)
2 O 1 M -> 1 O_2 1 M
1 O 1 O_2 1 N_2 -> 1 O_3 1 N_2
1 O 1 O_2 1 CO_2 -> 1 O_3 1 CO_2
1 O 1 O_3 -> 2 O_2
1 O 1 CO 1 M -> 1 CO_2 1 M
1 O(^1D) 1 O_2 -> 1 O 1 O_2
1 O(^1D) 1 O_3 -> 2 O_2
1 O(^1D) 1 O_3 -> 1 O_2 2 O
1 O(^1D) 1 H_2 -> 1 H 1 OH
1 O(^1D) 1 CO_2 -> 1 O 1 CO_2
1 O(^1D) 1 H_2O -> 2 OH
2 H 1 M -> 1 H_2 1 M
1 H 1 O_2 1 M -> 1 HO_2 1 M
1 H 1 O_3 -> 1 OH 1 O_2
1 H 1 HO_2 -> 2 OH
1 H 1 HO_2 -> 1 H_2 1 O_2
1 H 1 HO_2 -> 1 H_2O 1 O
1 O 1 H_2 -> 1 OH 1 H
1 O 1 OH -> 1 O_2 1 H
1 O 1 HO_2 -> 1 OH 1 O_2
1 O 1 H_2O_2 -> 1 OH 1 HO_2
2 OH -> 1 H_2O 1 O
2 OH 1 M -> 1 H_2O_2 1 M
1 OH 1 O_3 -> 1 HO_2 1 O_2
1 OH 1 H_2 -> 1 H_2O 1 H
1 OH 1 HO_2 -> 1 H_2O 1 O_2
1 OH 1 H_2O_2 -> 1 H_2O 1 HO_2
1 OH 1 CO -> 1 CO_2 1 H
1 HO_2 1 O_3 -> 1 OH 2 O_2
2 HO_2 -> 1 H_2O_2 1 O_2
2 HO_2 1 M -> 1 H_2O_2 1 O_2 1 M
1 N_2 -> 2 N
1 N_2 -> 2 N(^2D)
1 NO 1 hv -> 1 N 1 O
1 NO_2 1 hv -> 1 NO 1 O
1 NO_3 1 hv -> 1 NO_2 1 O
1 NO_3 1 hv -> 1 NO 1 O_2
1 N_2O 1 hv -> 1 N_2 1 O(^1D)
```

```
1 N_2O_5 1 hv -> 1 NO_2 1 NO_3
1 HNO_2 1 hv -> 1 OH 1 NO
1 HNO_3 1 hv -> 1 NO_2 1 OH
1 HO_2NO_2 1 hv -> 1 HO_2 1 NO_2
1 N 1 O_2 -> 1 NO 1 O
1 N 1 O_3 -> 1 NO 1 O_2
1 N 1 OH -> 1 NO 1 H
1 N 1 HO_2 -> 1 NO 1 OH
1 N 1 NO -> 1 N_2 1 O
1 N 1 NO_2 -> 1 N_2O 1 O
1 N(^2D) 1 O -> 1 N 1 O
1 N(^2D) 1 CO_2 -> 1 NO 1 CO
1 N(^2D) 1 N_2 -> 1 N 1 N_2
1 N(^2D) 1 NO -> 1 N_2 1 O
1 O 1 NO 1 M -> 1 NO_2 1 M
1 O 1 NO_2 -> 1 NO 1 O_2
1 O 1 NO_2 1 M -> 1 NO_3 1 M
1 O 1 NO_3 -> 1 O_2 1 NO_2
1 O 1 HO_2NO_2 -> 1 OH 1 NO_2 1 O_2
1 O(^1D) 1 N_2 -> 1 O 1 N_2
1 O(^1D) 1 N_2 1 M -> 1 N_2O 1 M
1 O(^1D) 1 N_2O -> 2 NO
1 O(^1D) 1 N_2O -> 1 N_2 1 O_2
1 NO 1 O_3 -> 1 NO_2 1 O_2
1 NO 1 HO_2 -> 1 NO_2 1 OH
1 NO 1 NO_3 -> 2 NO_2
1 H 1 NO_2 -> 1 OH 1 NO
1 H 1 NO_3 -> 1 OH 1 NO_2
1 OH 1 NO 1 M -> 1 HNO_2 1 M
1 OH 1 NO_2 1 M -> 1 HNO_3 1 M
1 OH 1 NO_3 -> 1 HO_2 1 NO_2
1 OH 1 HNO_2 -> 1 H_2O 1 NO_2
1 OH 1 HNO_3 -> 1 H_2O 1 NO_3
1 OH 1 HO_2NO_2 -> 1 H_2O 1 NO_2 1 O_2
1 HO_2 1 NO_2 1 M -> 1 HO_2NO_2 1 M
1 HO_2 1 NO_3 -> 1 O_2 1 HNO_3
1 NO_2 1 O_3 -> 1 NO_3 1 O_2
1 NO_2 1 NO_3 1 M -> 1 N_2O_5 1 M
1 NO_2 1 NO_3 -> 1 NO 1 NO_2 1 O_2
1 O 1 hv -> 1 O^+ 1 e
1 O_2 1 hv -> 1 O_2^+ 1 e
1 CO_2 1 hv -> 1 CO_2^+ 1 e
1 CO_2 1 hv -> 1 CO 1 O^+ 1 e
1 O_2^+ 1 e -> 2 O
1 CO_2^+ 1 e -> 1 CO 1 O
1 O^+ 1 CO_2 -> 1 O_2^+ 1 CO
1 O 1 CO_2^+ -> 1 O_2^+ 1 CO
1 O 1 CO_2^+ -> 1 O^+ 1 CO_2
1 CO_2^+ 1 H_2 -> 1 CO_2H^+ 1 H
1 CO_2H^+ 1 e -> 1 CO_2 1 H
1 HO_2 1 grain -> 1 (HO_2)_grain
1 (HO_2)_grain 1 OH -> 1 H_2O 1 O_2
```

E 9 Signal transduction and metabolic network

The signal transduction network has been obtained from the Reactome database (identifier: `REACT_111102.2`, www.reactome.org). The metabolic network has been obtained from the KEGG REACTION database (www.genome.jp/kegg). Both network models are to big to be printed here, but are contained on the supplementary CD.

Ehrenwörtliche Erklärung

Hiermit erkläre ich,

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

Bei der Auswahl und Auswertung des Materials haben mich folgende Personen unterstützt: PD Dr. Peter Dittrich und PD Dr. Stefan Artmann.

Ich habe die gleiche, eine in wesentlichen Teilen ähnliche bzw. eine andere Abhandlung nicht bei einer anderen Hochschule als Dissertation eingereicht.

Jena, den 31. Juli 2012