FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Algorithm for Computing all Persistent Subspaces of Reaction-Diffusion PDEs

Master Thesis
submitted in partial fulfillment of the requirement for the degree
Master of Science (M. Sc.)
in Bioinformatics

FRIEDRICH SCHILLER UNIVERSITY JENA
Faculty of Mathematics and Computer Science

submitted by

Linus Woitke

born 9.10.1997 in Altenburg

Jena, 28.02.2023

supervisor: Prof. Dr. Peter Dittrich and Dr. Stephan Peter

Zusammenfassung

Diese Arbeit stellt einen Algorithmus vor, der alle möglichen Unterräume für ein partielles Reaktions-Diffusions-Differentialgleichungssystem berechnet, die eine partielle Lösung der Gleichung enthalten können. Dazu werden alle möglichen Unterräume des zugrundeliegenden Reaktionsnetzes, die Distributed Organisationen (DOs) sind, identifiziert, da kürzlich gezeigt wurde, dass ein persistenter Unterraum eine DO sein muss. Der Algorithmus berechnet die Hierarchie der DOs und beginn dabei mit der größten, bis alle gefunden wurden. Der Ansatz ist dabei die lineare Programmierung (LP) mit Hilfe von integer cuts. Die zugrundeliegenden Constraints verwenden elementary reaction closure (ERCs) als minimale Bausteine, um lokale closedness und globale self-maintenance zu garantieren, die für eine DO erforderlich sind. Zusätzlich liefert der Algorithmus für jeden Unterraum und eine Menge aktiver Reaktionen (SAR) eine minimale Kompartimentierung, die notwendig ist, damit dieser Unterraum bestehen kann. Da die DOs ein Lattice bilden, ergibt sich eine hierarchische Struktur aller persistenten Unterräume des Reaktions-Diffusions-PDE-Systems. Der Algorithmus wird als Python-Quellcode zur Verfügung gestellt. Es wird gezeigt, dass der Algorithmus eine exponentielle obere Grenze der Laufzeitkomplexität hat. Danach wird der Algorithmus auf die BioModels-Datenbank angewendet, um Informationen aus bereits vorgestellten Modellen zu extrahieren. Dort werden Aussagen über den Einfluss von Zuflüssen und reversiblen Reaktionen gemacht. Neben den praktischen Implikationen des Algorithmus geben die Ergebnisse auch Einblicke in die Komplexität der Lösung von Reaktions-Diffusions-PDEs

Abstract

This thesis presents an algorithm computing all possible subspaces for a reaction-diffusion partial differential equation system, that can hold a persistent solution of the equation. For this, all possible sub-networks of the underlying reaction network that are distributed organizations (DOs) are identified, because recently it has been shown that a persistent subspace must be a DO. The algorithm computes the hierarchy of DOs starting from the largest by a linear programming (LP) approach using integer cuts. The underlying constraints use elementary reaction closures (ERCs) as minimal building blocks to guarantee local closedness and global self-maintenance, required for a DO. Additionally, the algorithm delivers for each subspace and a set of active reactions (SAR) a minimal compartmentalization that is necessary for this subspace to persist. Because DOs form a lattice, a hierarchical structure of all the persistent subspaces of the reaction-diffusion PDE system is obtained. The algorithm is provided as a python source code. It is shown that the algorithm has an exponential upper bound run time complexity. The algorithm is then applied to the BioModels database, to extract information from already introduced models. There statements are made regarding the impact of inflow and reversible reactions. Apart from the practical implications of the algorithm, the results also give insights into the complexity of solving reaction-diffusion PDEs.

Contents

1 Introduction **7**

2 Preliminaries **9**

3 Results **13**

3.1 Theoretical Results . . . 13
3.1.1 Set of Active Reactions (SARs) . . . 13
3.1.2 Maximal Reactive Compartments (MRCs) . . . 19
3.2 Algorithms . . . 20
3.2.1 Data Flow of Functions . . . 20
3.2.2 Function get_reactions() . . . 20
3.2.3 Function change_network() / generate_network_to_be_analyzed() . . . 21
3.2.4 Function create_ERCs() . . . 22
3.2.5 Solving for SARs (setup_LP(SAR)) . . . 23
3.2.6 Alternative Solver . . . 24
3.2.7 Solving for DOs (setup_LP(DO)) . . . 25
3.2.8 Function create_MRCs() . . . 25
3.2.9 Function get_minimal_compartments() . . . 25
3.2.10 Function get_Lattice() . . . 26
3.2.11 Analysis Class . . . 27
3.2.12 Function setup_LP_DOs_with_SAR() . . . 27
3.3 Runtime Analyses . . . 27
3.3.1 Termination . . . 27
3.3.2 Time Complexity remarks . . . 28
3.3.3 Time Complexity of getReaction() . . . 28
3.3.4 Time Complexity of change_network() . . . 28
3.3.5 Time Complexity of create_ERCs() . . . 29
3.3.6 Time Complexity of setup_LP() . . . 29
3.3.7 Time Complexity of get_lattice() . . . 30
3.3.8 Time Complexity of create_MRCs() . . . 31
3.3.9 Time Complexity of get_minimal_compartments() . . . 32

4 Examples and Evaluation **32**

4.1 Interpreting the Lattice of SARs . . . 32
4.2 Bio Models Database . . . 34
4.2.1 Approach . . . 34
4.2.2 Correctness of Os . . . 36
4.2.3 Handling Inflow Reactions . . . 38
4.2.4 Handling reversible Reactions . . . 38
4.2.5 Results of the Database . . . 40
4.2.6 Further Examination . . . 41
4.2.7 Separation of Support . . . 42
4.2.8 MRCs . . . 42
4.2.9 Compartmentalizations . . . 43

4.3 Randomly Generated Networks 44

5 Conclusion **44**

6 Bibliography **47**

7 Appendix **50**

8 Selbstständigkeitserklärung **51**

1 Introduction

Understanding a system and predicting its behavior is in general a complex problem because a system's behavior as a whole results in general from many non-linear interactions of its components. A particular challenge is to infer a system's behavior from its structure. In this work the structure of a system is described as a reaction network, that is a set of reaction rules over a set of molecular species (Aris 1965). Reaction network models are not only used in chemistry, but also in various other disciplines such as biology (Stephan Peter, Peter Dittrich, and Bashar Ibrahim 2021), physics (Jiang et al. 2021), computer science (Petri 1962), economy (Peter Dittrich and Winter 2005), or even social sciences (Peter Dittrich and Winter 2008).

Theoretical frameworks to analyze reaction networks differ in their focus, mathematical background, complexity, and their underlying assumptions. A number of approaches apply graph theory while ignoring mass conservation and stoichiometric relationships of the reaction rules. For example, these approaches simply check for the connectivity of molecular species (Figueiredo et al. 2008) or estimate the node degree distribution to identify small-world characteristics (Wagner and Fell 2001).

However, behavior prediction is very limited due to the level of abstraction, as a reaction network can mathematically be seen as a hyper-graph, which contains more information than a simple graph of species (Figueiredo et al. 2008).

An approach respecting all stoichiometric relationships of the reaction network is chemical reaction network theory, which has been introduced in between 1960-1970 through the works by Aris (1965), Horn and Jackson (1972) and Feinberg (1972). This theory focuses on a generalization of a reaction network structure and the variety of dynamics this model might express as a system of differential equations. A key point of this theory is that qualitative dynamic behaviors are linked to the reaction network's structure. For example, the deficiency theorems by Feinberg (1987) provide structural prerequisites for an asymptotically steady state; and Dickenstein et al. (2019) study conditions for multistationarity, that is, two or more positive steady states with the same conserved quantities. They use degree theory to check whether "critical functions" change their sign.

A similar approach introduced by Gatermann (2001) uses real valued algebraic geometry and discrete mathematics to link a network to qualitative dynamical properties like the number of stable states (Gatermann 2001), Hopf-bifurcations (Gatermann, Eiswirth, and Sensse 2005) and chaos (Sensse and Eiswirth 2005).

Such a reduction of complexity, by being able to represent dynamics with a smaller number of parameters is also done by Cardelli (2014), who implemented morphisms of reaction networks. This points out similarities in structure between reaction networks of different sizes.

This thesis follows a different approach called chemical organization theory (COT) (Fontana and Buss 1994; Peter Dittrich and Di Fenizio 2007), which does not assume particular kinetic laws like mass-action kinetics and which has been developed to tackle the chemical evolution of systems consisting of a huge, even infinite, number of molecular

species. The basic idea is to identify a chemical organization, which is a set of species that are closed and self-maintaining. It has been proven that these organizations are linked to the long-term behaviour of the reaction network (Peter Dittrich and Di Fenizio 2007; S. Peter and P. Dittrich 2011).

There are several algorithms to attain the organizations of a (finite) reaction network. One algorithm computes all semi-organizations and then checks all semi-organizations for self-maintainence using linear programming. A semi-organization is a closed species set that produces each of its consumed species. The semi-organizations are built up iteratively from smaller ones by adding species. A second approach is a flux based approach, where self-maintaining flux distributions (elementary modes) are combined to discover closed sets (Centler, Kaleta, Fenizio, et al. 2008). Note that the question if a reaction network has a reactive organization apart from the empty set is NP-complete (Centler, Kaleta, Fenizio, et al. 2008) .

Centler, Kaleta, Fenizio, et al. (2008) showcased the performance of these algorithms for models of different sizes. Whereas the first is more certain to terminate with all solutions found, the second has a reduced computational time for organizations sets that have combinatorial amounts. Centler, Kaleta, Speroni di Fenizio, et al. (2010) extended the algorithm by applying multiprocessing for the steps of semi-organizations as well as the part of linear programming to ensure self-maintenance.

The basic theory does not consider a spatial or temporal distribution of the system's components. However when looking at organisms, for example the eukaryotic cell, we observe spatial arrangements. Different parts of the cell perform different reactions to create an overall cycle attaining self-maintenance. The need for spatial separation and specialization forms the basis of multicellular organisms. On the other hand, there is also temporal separation that is not as easy to grasp but can be observed in biological systems, e.g. the day-night rhythm of plant respiration. Reaction network models for this phenomena exhibit periodical behavior.

To link spatial and temporal separation with traditional organization theory, Stephan Peter, Bashar Ibrahim, and Peter Dittrich (2021) recently suggested the concept of distributed organizations. This is a set of species that achieves overall self-maintenance by separation into suitable compartments (Stephan Peter, Bashar Ibrahim, and Peter Dittrich 2021). The mathematical framework combines self-maintenance and stability, closure and distribution of species and can make statements about complex systems that are influenced by spatial properties to ensure their survival as a whole. The main result of the paper is theorem 3.4.1., which states that a set of persistent species has to be DO in a bounded system. There they make an extension to the concept of persistence, that does not ask for a species to be greater than the constant ϵ for time towards infinity, but also allows the species to be smaller than this constant, if it surpasses this threshold later on. This indicates DOs as possible prestep towards the identification of all persistent solutions of a reaction diffusion PDE. While a mathematical theory was introduced, there was no algorithm available so far.

In 2020 a first version of an algorithm was released in the context of a bachelor thesis by the author, which is accessible through github `https://github.com/WoitkeL/`

`algorithm_DO`. This algorithm shares the idea of the build up through the smallest reaction closures and the general build up from SBML files. It differs by only focusing on the level of species as it can only solve for feasible DOs. It also lacks an effective reduction of the ERCs as well as the concept of MRCs, which both will be explained later on. For this reason, the minimal number of compartments is a simple heuristic that is driven by merges of species subsets. It is also inferior in terms of computation time since it lacks a faster solver.

The goal of this thesis is to extend these algorithms and supply a computationally effective objects and perform solid analysis for the applicability and provide a user friendly interface with functions to gain information towards distributions of networks. The computational complexity of the algorithms will be evaluated to ensure there appliance to models. In developing the more sophisticated algorithms, it was determined that it is easier to look for the different possible reaction vectors, instead of exploring each possible subspace of the model. By changing the view in this way, no relevant information is lost.

The presented algorithm practically creates a compartment for each reaction that is active, resulting in a loss of complexity and introducing the option to solve the system with constraint based linear programming. This means that even large reaction networks with up to thousands of reactions can be solved. Moreover models where the algorithms for organizations reach their limits can be calculated. The lost complexity from skipping the search of efficient compartments that execute multiple reactions, is addressed in a second part of the algorithm. There the minimal number of compartments needed for a set of reactions by finding a subset of promising compartments is determined. After that an ILP is used again, to determine the minimal set cover. The fundamental new concepts of an elementary reaction closure (ERC) and maximal reactive compartment (MRC) will be explained and put into biological context. These concepts will also be placed next to the already established mathematical concepts of the distributed organizations. The evaluation is done by comparing the results with the results of the paper by Kaleta, Richter, and Peter Dittrich (2009). The algorithm is then applied to already introduced models of the BioModels database. An in-depth analysis addresses the results.

2 Preliminaries

In this section the definitions and main results from (Stephan Peter, Bashar Ibrahim, and Peter Dittrich 2021) are introduced. The following definitions are mainly transcribed with their consent. A reaction network $(\mathcal{S}, \mathcal{R})$ consists of a finite set $\mathcal{S}$ of $n \in \mathbb{N}$ species as well as a finite set $\mathcal{R}$ of $m \in \mathbb{N}$ reactions describing the interactions between the species. As an example, we consider the following reaction network from the role of microRNAs in osteoarthritis by Proctor and Smith (2017). This is a micro-RNA transcription-factor

interaction model: The set of species is

$$\begin{aligned} TF1 &\to \text{Transcription factor 1} \\ TF2 &\to \text{Transcription factor for miR synthesis} \\ miR &\to \text{micro RNA} \\ miR\_gene &\to \text{gene of micro RNA} \\ Sink &\to \text{EmptySet} \\ Signal &\to \text{signal of TF1 transcription} \\ miR\_gene\_TF2 &\to \text{mir\_gene\_TF2 complex} \\ miR\_gene\_TF1 &\to \text{mir\_gene\_TF1 complex} \\ TF1\_mRNA &\to \text{TF1\_mRNA complex} \end{aligned}$$

and the set of reactions

$$\begin{aligned} r_1 &: miR\_gene + TF1 \longrightarrow miR\_gene\_TF1 && \text{miR-gene TF1 binding} \\ r_2 &: miR\_gene\_TF1 \longrightarrow miR\_gene + TF1 && \text{miR-gene TF1 release} \\ r_3 &: miR\_gene + TF2 \longrightarrow miR\_gene\_TF2 && \text{miR-gene TF2 binding} \\ r_4 &: miR\_gene\_TF2 \longrightarrow miR\_gene + TF2 && \text{miR-gene TF2 release} \\ r_5 &: miR\_gene\_TF2 \longrightarrow miR\_gene\_TF2 + miR && \text{miR-synthesis} \\ r_6 &: miR \longrightarrow Sink && \text{miR-degradation} \\ r_7 &: Signal \longrightarrow Signal + TF1\_mRNA && \text{TF1-transcription} \\ r_8 &: TF1\_mRNA \longrightarrow Sink && \text{TF1-mRNA-degradation} \\ r_9 &: TF1\_mRNA + miR \longrightarrow miR && \text{TF1-mRNA-deg via miR} \\ r_{10} &: TF1\_mRNA \longrightarrow TF1\_mRNA + TF1 && \text{TF1-translation} \\ r_{11} &: TF1 \longrightarrow Sink && \text{TF1-degradation,} \end{aligned}$$

thus $n = 8$ and $m = 11$ for this example. Generally, the reaction equation for species number i can be described by

$$\sum_{j=1}^{m} a_{ij} \to b_{ij} \tag{1}$$

with natural numbers $a_{ij}, b_{ij},\ j = 1, \ldots, m$. For reaction number j we call the set of species s_i with $a_{ij} > 0$ the support of r_j, shortly $supp(r_j)$, and the set of species s_i with $b_{ij} > 0$ the products of r_j, shortly $products(r_j)$. From the set of reactions the so-called stoichiometric matrix $N \in \mathbb{R}^{n \times m}$ is derived.

Definition 1 (Closure of a subset of species) *Given a reaction network $(\mathcal{S}, \mathcal{R})$ and a subset $S \subseteq \mathcal{S}$ of species. We define the set operation*

$$clos_1(S) \equiv S \cup \{s_i \in \mathcal{S} : \exists r_j \in \mathcal{R} : \ supp(r_j) \subseteq S,\, b_{ij} > 0\}, \tag{2}$$

this describes a set of species $s \in \mathcal{S}$ as well as all species that are the product of the reactions supported by s. It can be represented by the repetitive formation of closure, resulting in a monotonously increasing sequence of sets

$$\begin{aligned} clos_1^0(S) &= S, \\ clos_1^1(S) &= clos_1(S), \\ clos_1^2(S) &= clos_1(clos_1(S)), \\ clos_1^3(S) &= clos_1(clos_1(clos_1(S))), \\ &\dots \\ clos_1^{k_{min}+1}(S) &= clos_1(clos_1^{k_{min}}(S)), \end{aligned}$$

where $k_{min} = \min\{k \in \mathbb{N}_0 : clos_1^{k+1}(S) = clos_1^k(S)\}$. Since the set of species and the set of reactions are finite, k_{min} is finite and thus the closure of S is unique and finite. We call the set

$$clos(S) \equiv clos_1^{k_{min}}(S) \tag{3}$$

the closure of S.

Generally, flux vectors $v \in \mathbb{R}_+^m \equiv \{x \in \mathbb{R} : \ x \geq 0\}$ are used in dynamical systems to describe the intensity of each reaction for a given state of the system. Depending on the present species in that state not all flux vectors are feasible ones, because a reaction is active if and only if all the species of its support are present. For a given subset $S \subseteqq \mathcal{S}$ of species, a vector $v \in \mathbb{R}_+^m$ of m non-negative real numbers v_r is called *feasible flux* (with respect to S) if and only if for all reactions $r \in \mathcal{R}$

$$v_r > 0 \Leftrightarrow supp(r) \subseteqq S. \tag{4}$$

Furthermore a reaction $r \in \mathcal{R}$, that has a support of $\emptyset$ is called inflow reaction as the support of this reaction is always fulfilled. Such a reaction is to be seen as r1 in the presented reaction network

Definition 2 (Closedness, self-maintenance and organizations) *Given a reaction network $(\mathcal{S}, \mathcal{R})$ and a subset $S \subseteq \mathcal{S}$ of species then we call S*

1. *self-maintaining if there is a feasible flux v with respect to S such that*

$$N \cdot v \geq 0, \tag{5}$$

   that is, all elements of $N \cdot v$ are equal or greater than zero,

2. *closed if*

$$clos(S) = S, \tag{6}$$

3. *organization if it is self-maintaining and closed.*

The previously defined organizations are generalized towards so-called distributed organizations (DOs) in the next definition. DOs are introduced and broadly discussed in (Stephan Peter, Bashar Ibrahim, and Peter Dittrich 2021). In this paper an algorithm for the computation of DOs is presented and analyzed.

Definition 3 (Distributed organizations (DOs)) *Given a reaction network $(\mathcal{S}, \mathcal{R})$, a subset $D \subseteq \mathcal{S}$ is a DO (through $\hat{v} \in \mathbb{R}_+^m$) if and only if there are k, $k \in \mathbb{N}$, different subsets (that we call "compartments" according to nomenclature of systems biology) $S_1, \ldots, S_k \subseteq D$ with*

$$D = \cup_{i=1}^{k} S_i \tag{7}$$

such that

1. *all S_i, $i = 1, \ldots, k$, are closed;*
2. *there is a vector $\hat{v} \in \mathbb{R}_+^m$, $\hat{v} \geq 0$, such that*

$$N\hat{v} \geq 0; \tag{8}$$

3. *and there is a feasible flux $\hat{v}^i \in \mathbb{R}_+^m$, $\hat{v}^i \geq 0$, with respect to each subset S_i, $i = 1, \ldots, k$, with*

$$\hat{v} = \sum_{i=1}^{k} \hat{v}^i. \tag{9}$$

Collectively, we call the equations (8) and (9) the self-maintenance property of a DO. We say "D is distributed to the compartments S_i" or "the S_i are a distribution of D". When listing the elements of the subsets S_i, $i = 1, \ldots, k$, of species, a special notation is used, for example, if D is distributed to $S_1 = \{s_1, s_2\}$ and $S_2 = \{s_1, s_3\}$, we write

$$D = S_1 \cup S_2 = \{s_1 s_2 | s_1 s_3\}. \tag{10}$$

If a DO exhibits a distribution to only one subset of species, then this DO is an organization in the sense of COT.

"Mathematically, the significance of DOs is proven by the fact that the set of persistent species for every solution of a reaction-diffusion system is always a DO." (Stephan Peter, Bashar Ibrahim, and Peter Dittrich 2021)

From a given reaction network, the DOs can be computed without the need of any knowledge about the kinetics (reaction constants, kinetic laws applied, etc.). The set of DOs is always a lattice (Stephan Peter, Bashar Ibrahim, and Peter Dittrich 2021). The lattice of DOs for the example, is shown in Figure 1.

The left-hand side of Figure 2 shows the main definitions of this subsection together with their realations to one another.

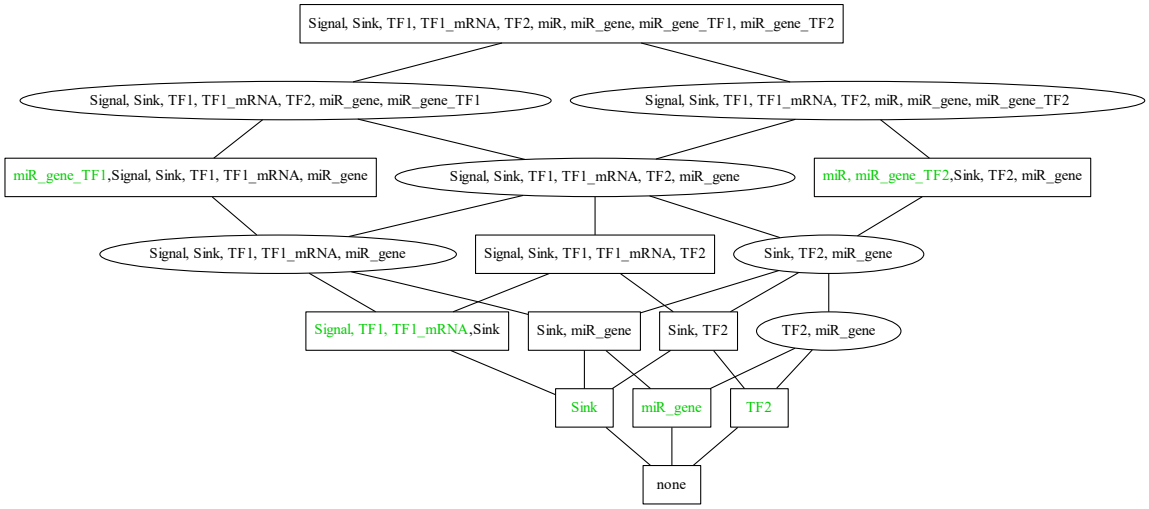


Figure 1: Lattice of DOs ((Stephan Peter, Peter Dittrich, and Bashar Ibrahim 2021)) of micro-RNA transcription-factor interaction model from Proctor and Smith (2017). Overall there are 17 DOs. The boxes mark DOs, that are organizations, while the 6 oval DOs are not an organization. The ladder species sets can therefore not exist in one compartment and have to be distributed to satisfy the attributes of a DO. Each DO indicates its species. Species that do not appear in a subset of the DO are marked green. The smallest organization is at the bottom of the lattice and contains the empty set as species set. There is no information about the possible active reactions in the DO. The species $TF2, miR\_gene$ and $Sink$ do not trigger any reactions on their own and therefore create multiple DOs that are non-reactive like the empty set. At the top of the lattice is the biggest DO. Here it contains all species of the model.

3 Results

3.1 Theoretical Results

In this section the new definitions which are necessary to formulate our algorithm, are stated.

3.1.1 Set of Active Reactions (SARs)

In principal, this is a transfer of the species-based definitions from above to a reaction-based approach. The first idea, is to transfer the definition 1 of the closure of a subset of species to reactions.

Definition 4 (Elementary Reaction Closure (ERC)) *Given a reaction network* $(\mathcal{S}, \mathcal{R})$

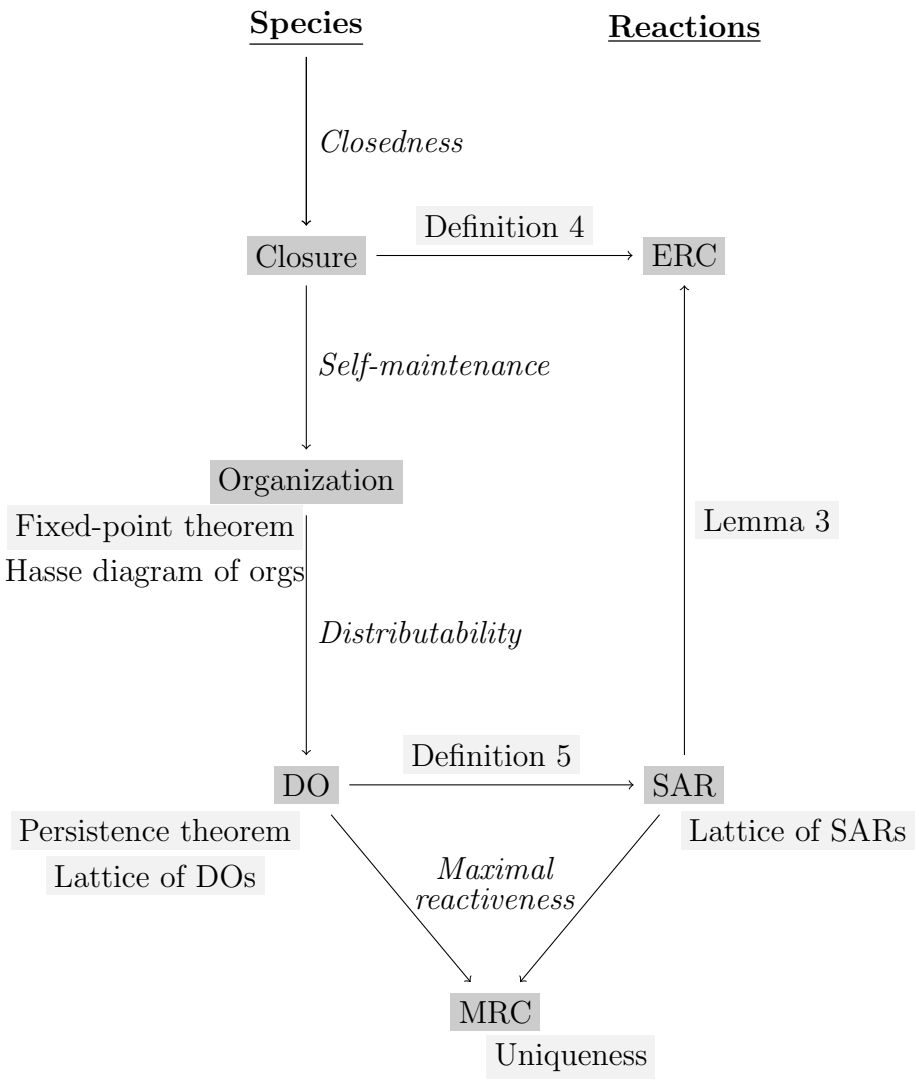


Figure 2: Overview of the relation between the main definitions.

and a reaction $\hat{r} \in \mathcal{R}$*. We call the set*

$$ERC(\hat{r}) \equiv \{r \in \mathcal{R} : \, supp(r) \subseteq clos(supp(\hat{r}))\} \tag{11}$$

of reactions the elementary reaction closure (ERC) of $\hat{r}$*.*

By design, the ERC of a reaction is unique. In the implementations of the algorithm, the computation of the ERC of a reaction is realized by the function $create\_ERCs()$. For the example of the micro-RNA transcription-factor interaction model, Table 1 shows the ERCs of each reaction. The two ERCs of r_1 and r_6 of the table are explained, to exemplify how ERCs can be constructed iteratively. To run the reaction r_1 the species of $miR_gene, TF1$ are required to support the reaction. The reaction r_1 produces the species miR_gene_TF1 only. The species set $miR\_gene, TF1, miR\_gene\_TF1$ contains the support of the reaction $r2$ that is then added to the ERC. The products of this reaction to not expand the species set of the ERC. The set also contains the support of the reaction r_{11}. This reaction expands the sepcies set to $miR\_gene, TF1, miR\_gene\_TF1, Sink$. There are no further reactions

Reactions	ERCs
r_1	r_1, r_2, r_{11}
r_2	r_2, r_1, r_{11}
r_3	r_3, r_4, r_5, r_6
r_4	r_4, r_3, r_5, r_6
r_5	r_5, r_4, r_6, r_2
r_6	r_6
r_7	r_7, r_8, r_{10}, r_{11}
r_8	r_8, r_{10}, r_{11}
r_9	r_9,r_6,r_8, r_{10}, r_{11}
r_{10}	r_{10}, r_8, r_{11}
r_{11}	r_{11}

Table 1: ERCs for all reactions of the micro-RNA transcription-factor interaction model from Proctor and Smith (2017).

which are supported. This means the ERC of $\{r_1$ is $r_1, r_2, r_{11}\}$ The reaction r_6 has miR as support and $Sink$ as product. This set of species does not support any other reactions. Therefore the ERC of r_6 is $\{r_6\}$

Definition 5 transfers DOs, which were defined for species, towards sets of active reactions, which are based upon reactions.

Definition 5 (Set of Active Reactions (SAR) and Overproduction) *Given a reaction network $(\mathcal{S}, \mathcal{R})$, a DO $D \subseteq \mathcal{S}$, and a vector $\hat{v} \in \mathbb{R}^m_+$, such that D is a DO through $\hat{v}$, then we say that*

$$SAR(\hat{v}) \equiv \{r_j \in \mathcal{R} : \hat{v}_j > 0\} \tag{12}$$

is a set of active reactions (through $\hat{v}$).
Furthermore a species $s_i \in \mathcal{S}$ is overproduced with respect to the flux vector $\hat{v}$ if and only if

$$(N\hat{v})_i > 0. \tag{13}$$

Similar to the concept of compartments in regard to DOs, an annotation of the inner structure of SARs by compartments is defined. For a SAR, we say that the SAR has a valid distribution to the compartments of $S_1, \ldots, S_j$ with an affiliated set of reactions $R(S_1), \ldots, R(S_j)$ with

$$R(S_i) = \{r : supp(r) \in S_i \text{ for } r \in \mathcal{R}\}, \tag{14}$$

shortly

$$SAR = R(S_1) \cup \ldots \cup R(S_j), \tag{15}$$
$$S(SAR) = S_1 \cup \ldots \cup S_j, \tag{16}$$

where

$$S(SAR) = \bigcup_{r \in SAR} (supp(r) \cup prod(r)) \tag{17}$$

is the set of all species involved in reactions from the SAR. Thus, $S(SAR)$ is the smallest DO that can support the SAR.

The lattice of SARs for the example, is shown in Figure 3.

Lemma 1 (Unique Set of Overproduction) *To each $SAR(\hat{v})$ belongs a unique biggest set of species that can be overproduced by this set of reactions.*

Proof: There can be a number of flux vectors $\hat{v}_1, \ldots, \hat{v}_k \in \mathbb{R}^m$, tracing to the same SAR with $SAR = \{r_j \in \mathcal{R} : \hat{v}_j > 0\}$, but with different species of overproduction. Through union of all possible vectors that are not a multiple of one another, we gain a unique biggest set of overproduced species.

The following lemma describes the relation between DOs and SARs.

Lemma 2 (Relation between DOs and SARs) *For a given reaction network $(\mathcal{S}, \mathcal{R})$*

1. *there can be several flux vectors $\hat{v}_1, \ldots, \hat{v}_k \in \mathbb{R}^m_+$, through which a subset $D \subseteqq \mathcal{S}$ of species is a DO, such that the $SAR(\hat{v}_1), \ldots, SAR(\hat{v}_k)$ are possibly different from each other. These differences describe the different potential behaviors of the DO D in terms of the active reactions. It is to note, that the flux vectors are complete for union, but they are not guaranteed to have a unique supremum and therefore do not resemble a lattice.*
2. *on the other hand, for a given a SAR $S \subseteqq \mathcal{R}$ there can be different DOs that can perform this SAR through compartmentalization. These DOs consist of a minimal DO D as well as DOs consisting of D as well as a number of non-reactive reactants. Since these non-reactive reactants can exist in a separate compartment from one another, the possible DOs that can perform a certain SAR build up a lattice as well. This attribute is used later on, to calculate all DOs of a system, by analyzing the their SARs.*

These relations are shown on the lattices of SARs and DOs of the example. The first is the mapping of DOs to SARs. In the lattice we do not see which SAR the DOs can accomplish. We can have DOs, that can perform several SARs, with different compartmentalizations: for example the largest DO {*Signal*, *Sink*, $TF1$, $TF1\_mRNA$, $TF2$, *miR*, *miR_gene*, *miR_gene*_$TF1$, *miR_gene*_$TF2$} is able to perform the largest SAR as well as the second largest SAR (connected through red line). Normally, it is common that these DOs are associated with multiple SARs, but this is reduced by species that are the support of first-order reactions. Also this SAR is only possible in the largest DO. This means that when optimizing for the largest set of reactions that a DO can perform, we do not see this possible SAR in the lattice of DOs.

Lemma 3 provides an equivalent definition of SAR that uses ERCs and is used for the implementation of the algorithm.

Lemma 3 (Equivalent Definition for SARs) *Given a reaction network $(\mathcal{S}, \mathcal{R})$, a vector $\hat{v} \in \mathcal{R}_+^m$ with $N \cdot \hat{v} \geq 0$, and $\hat{R} \equiv \{r_j \in \mathcal{R} : \hat{v}_j > 0\}$, then the following two statements are equivalent:*

1. *There is a subset $D \subsetneqq S$ which is a DO through $\hat{v}$, that is, $\hat{R}$ is a set of active reactions through $\hat{v}$.*
2. *$ERC(r) \subsetneqq \hat{R}$ for all $r \in \hat{R}$.*

Proof: 1. $\Rightarrow$ 2. :
We assume that statement 1. is true and choose arbitrary reactions $\hat{r} \in \hat{R}$ and $r \in ERC(\hat{r})$. It is to show that $r \in \hat{R}$. Since $\hat{r} \in \hat{R}$ and D is a DO, there is a closed subset $\hat{S} \subsetneqq D$ with $supp(\hat{r}) \subsetneqq \hat{S}$ and a feasible flux $v \in \mathbb{R}_+^m$ for $\hat{S}$ with $v \leq \hat{v}$. Since $r \in ERC(\hat{r})$, it holds $supp(r) \subsetneqq clos(supp(\hat{r}))$. Together, we have

$$supp(r) \subsetneqq clos(supp(\hat{r})) \subsetneqq clos(\hat{S}) = \hat{S}. \tag{18}$$

(The last equation holds true, because $\hat{S}$ is closed.) It follows that $\hat{v}_r \geq v_r > 0$. This results in $r \in \hat{R}$ and thus $ERC(\hat{r}) \subsetneqq \hat{R}$, q.e.d.

2. $\Rightarrow$ 1. :
When assuming that statement 2. is true it is to show that there are closed subsets $S_1, \ldots, S_k \subsetneqq D$ together with a feasible flux $\hat{v}^i \in \mathbb{R}_+^m$ for each S_i, $i = 1, \ldots, k$, such that $\sum_{i=1}^k \hat{v}^i = \hat{v}$. Let $k \equiv |\hat{R}|$, that is, $\hat{R} = \{r_1, \ldots, r_k\}$. For $i = 1, \ldots, k$, we define closed subsets $S_i \equiv clos(supp(r_i))$ of species. For each of these subsets and each reaction $r_j \in \mathcal{R}$, $j = 1, \ldots, k$, we define the numbers

$$l(j) \equiv \{i \in \{1, \ldots, k\} : supp(r_j) \subsetneqq S_i\}, \tag{19}$$

which is the number of reactions from $\hat{R}$ with an ERC supporting r_j, and

$$l(i,j) = \begin{cases} 0, & \text{if } supp(r_j) \not\subsetneqq S_i \\ 1, & \text{if } supp(r_j) \subsetneqq S_i, \end{cases} \tag{20}$$

which equals one, if r_j is supported by S_i, and 0, if r_j is not supported by S_i. Now, for each subset S_i, $i = 1, \ldots, k$, and each reaction $r_j \in \mathcal{R}$, $j = 1, \ldots, m$, we define

$$\hat{v}_j^i = \begin{cases} \dfrac{l(i,j)}{l(j)} \hat{v}_j, & \text{if } r_j \in \hat{R} \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Lemma 4 states that the set of all SARs forms a lattice.

Lemma 4 (SARs form a lattice) *The set of all SARs of a given reaction network $(\mathcal{S}, \mathcal{R})$ forms a lattice.*

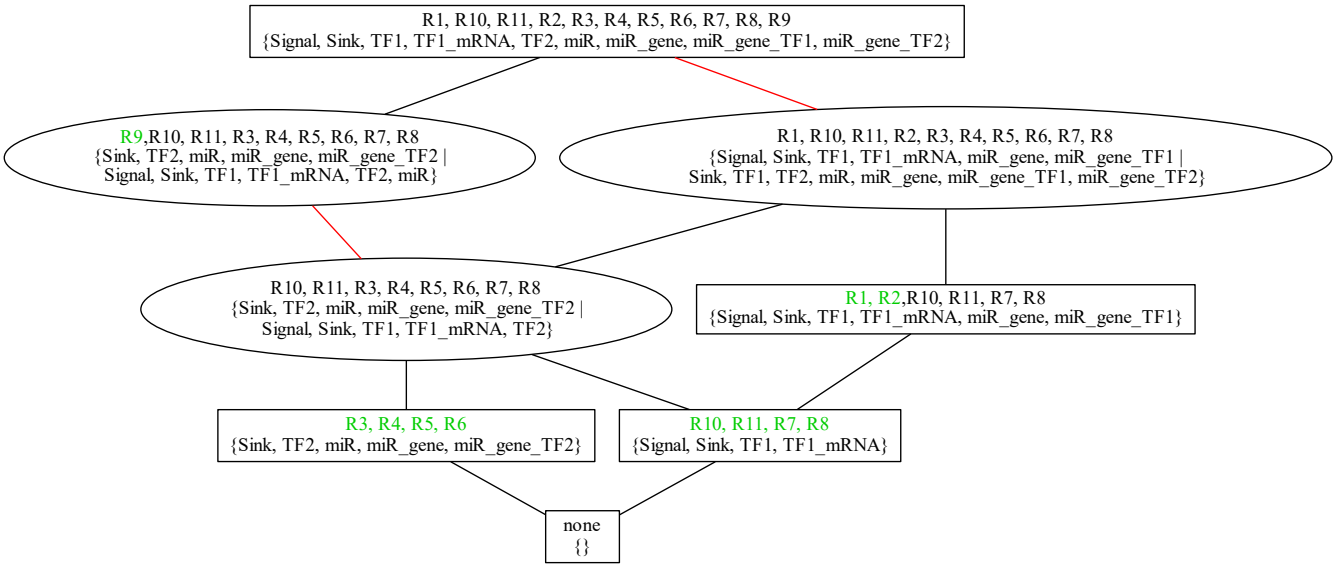


Figure 3: This lattice shows the 8 SARs any DO might be able to perform. Below the SAR, we see a possible compartmentalization to achieve the it. The boxes mark SARs that can occur in an O. The red lines indicate that 2 SARs have the same S(SAR). A more detailed description of all informations of the lattice is given in 4.1

.

Proof: A lattice is a partially ordered set in which every two elements have a unique supremum and a unique infimum.

1. *Partial order of the set of SARs*: The subset relation for sets provides a partial order.

2. *Unique supremum*: Given two SARs $R_1, R_2 \subseteq \mathcal{R}$, the set union of 2 SARs (put into different compartments of the union) forms a unique supremum

$$R_{sup} \equiv R_1 \cup R_2. \tag{22}$$

3. *Unique infimum*: Given two SARs $R_1, R_2 \subseteq \mathcal{R}$ of the reaction network we take the union of all SARs in $R_1 \cap R_2$ as infimum that is,

$$R_{inf} \equiv \cup\{R \subseteq R_1 \cap R_2 : \text{ R is a SAR}\}. \tag{23}$$

The existence of R_{inf} follows from the fact that there exists a unique minimal SAR R_{min} of the reaction network. This is the set of inflow reactions, as well as the reactions in the ERCs of the inflow reactions. The reactions of R_{min} are included in any SAR. Note that R_{min} can be empty.

3.1.2 Maximal Reactive Compartments (MRCs)

Obviously, the complexity of a DO increases with its number of species. Furthermore, the complexity of a distribution of a SAR increases with the number of its compartments. For real life systems evolutionary aspects such as efficiency are to consider, i.e., it is to assume that sustaining many compartments results in additional costs for the system. Because of this, the focus is on the distributions with the least number of compartments. A SAR can be performed by a finite amount of compartments, when we demand that the compartments are different to each other.

Definition 6 takes this into account and furthermore limits the computational expense. The minimal number of compartments of a SAR is unambiguous but has several optimal solutions.

Definition 6 (Maximal Reactive Compartments (MRC)) *Given a reaction network $(\mathcal{S}, \mathcal{R})$, closed subsets $S_1, \ldots, S_l \subseteqq \mathcal{S}$ of species, subsets $R_1, \ldots, R_l \subseteqq \hat{R}$ of associated reactions, and a SAR $\hat{R} = \{\hat{r}_1, \ldots, \hat{r}_k\} \subseteqq \mathcal{R}$ consisting of k reactions, such that*

$$\cup_{i=1}^{l} R_i = \hat{R} \tag{24}$$

and

$$supp(r) \subseteqq S_i. \tag{25}$$

all $r \in R_i$, $i = 1, \ldots, l$. Then we call the set of subsets $S_1, \ldots, S_l \subseteqq \mathcal{S}$ of species the maximal reactive compartments (MRC) of the SAR $\hat{R}$, shortly $MRC(\hat{R})$, if the following condition holds true for all $s \in \mathcal{S}$ and all $i = 1, \ldots, l$:

$$clos(S_i \cup \{s\}) \subseteqq \hat{R} \Leftrightarrow s \in S_i. \tag{26}$$

It follows that $S_i \not\subseteqq S_j$ for all $i, j \in \{1, \ldots, l\}, i \neq j$, that is, no species subset is a proper subset of another species subset.

The uniqueness of MRCs is proven by lemma 5.

Lemma 5 (Uniqueness of MRCs) *Given a SAR $\hat{R}$ as in Definition 6, its set of MRCs $\{S_1, \ldots, S_l\}$ is unique.*

Proof: Each $S \in \{S_1, \ldots, S_l\}$ exhibits the following properties:

- it does not support a reaction outside the SAR
- it is closed
- it is not a proper subset of another element of $\{S_1, \ldots, S_l\}$ with the previous two properties.

The properties mentioned above are unambiguous and thus the set of MRCs is unique.

The right-hand side of Figure 2 shows the new definitions of this subsection and relates them to those from the Preliminaries, which can be found on the left-hand side.

We can see the appliance of MRCs on the SAR $TF1\_degradation$, $TF1\_mRNA\_degradation$, $TF1\_transcription$, $TF1\_translation$, $miR\_degradation$, $miR\_gene\_TF2\_binding$, $miR\_gene\_TF2\_release$, $miR\_synthesis$. It can be seen as the vertex R10, R2, R3, R4, R5, R6, R7, R8, R9 in the lattice of SARs 3. The algorithm produces 3 MRCs:

- MRC 1: {Signal, Sink,TF2, TF1_mRNA,TF1}
- MRC 2: {Sink, miR_gene, TF2, miR_gene_TF2, miR}
- MRC 3: {Sink, TF2, miR, TF1}

The MRCs 1 and 2 are able to perform the SAR. Resulting in a minimal number of compartments of 2. These compartments are not unique. As we can see, one can remove the species $TF2$ from the MRC2 without impacting the active reactions. The reactions supported in a MRC can also be impacted by removing a species of its support, but only if the support of these reactions is also in another active compartment.

3.2 Algorithms

In this section all necessary algorithms are introduced.

3.2.1 Data Flow of Functions

In Figure 4 the specific functions are shown interacting as a workflow including the most important data objects. These functions can be used independently or handled by the analyze class. For a more detailed version that assigns the class functions respectively see the extended Figure available through GitHub.

Now single parts of the *pipeline_DO*() are addressed as given in figure 4.

3.2.2 Function get_reactions()

Input: path of SBML-file (default: built-in reaction network)
Result: list of objects (class Reaction) → list_of_all_reactions
Parameters: consider_reverse, consider_constant , consider_init_ammount, get_name
At first, the network of reactions has to be extracted from a SBML-document. For that it requires a path to a sbml- or xml-file. If none is given, the program will work with a default reaction network. This can be altered manually, if required.
The network is then saved as a list of single reactions. These reactions are objects of the implemented class *Reaction*. The attributes of this class manage reactants, products and the properties of closure in regard to the species set. This class uses the SBML reaction class to obtain its information.
Since this is the only interaction with the sbml-file, we have the option to extract the

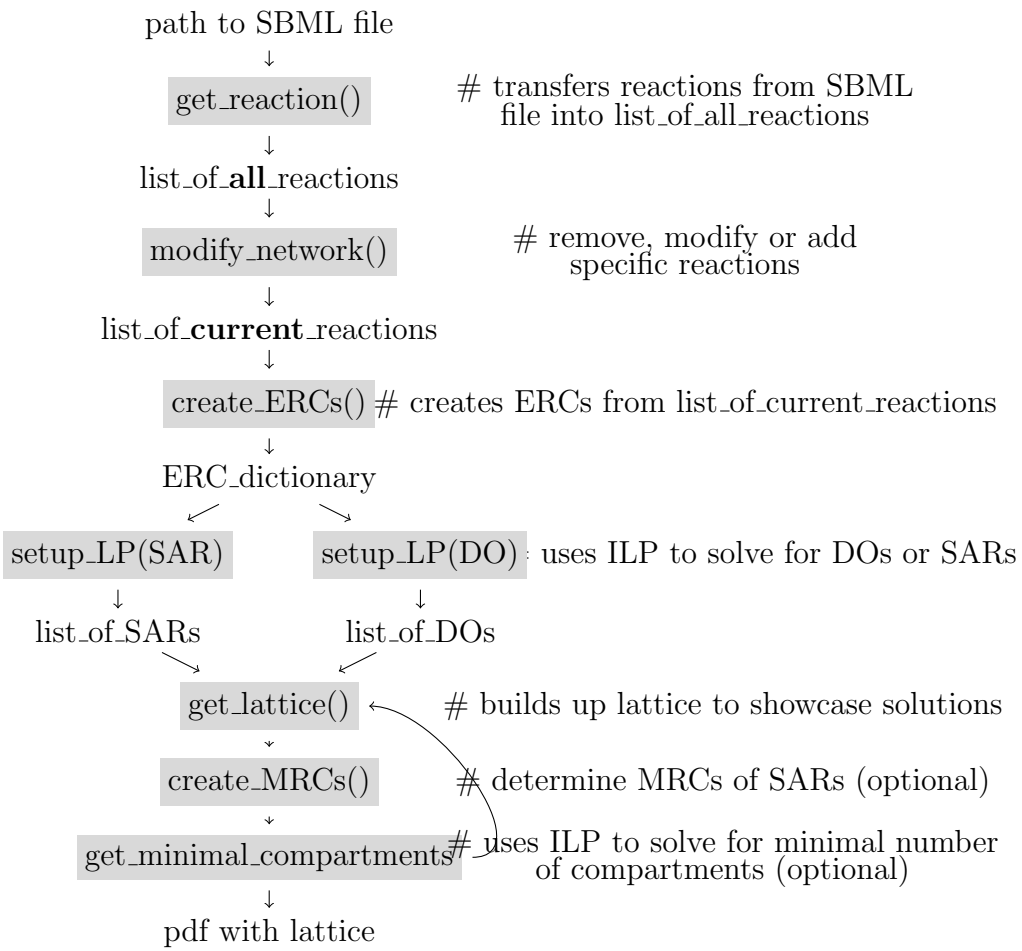


Figure 4: The functions (gray boxes) called by *pipeline_DO*() together with their respective inputs and outputs. ILP stands for integer linear programming. To get directed to the descriptions of the mentioned functions click on their respective boxes.

model name with the the parameter get_name=True which results in returning the modelname as second object. There are also 3 boolean parameters that add additional reactions to the reaction network for each object, where the criteria is met. These consist of: *consider_reverse*, to add a reversed reaction when the *reversible* attribute of a SBML reaction is set to *True*, or adding an inflow reaction if a species is set as *constant* or its *InitialAmount* is greater 0 for the parameters *consider_constant* and *consider_init_ammount*, respectively.

3.2.3 Function change_network() / generate_network_to_be_analyzed()

Input: list of objects (class Reaction)
Result: list of objects (class Reaction)
Parameters: species, add_reaction, exclude_species, make_inflow_seperable
This function interacts as an interface to make changes in the given SBML-Model. This is implemented by several parameters.
Species: allows to analyse a subset of species of the reaction network, by removing all

reactions that can not be supported in the subset to reduce computation. Also, any reaction which produces at least one element outside of the set, has to be inactive, since it otherwise violates the borders of the species set. For these reactions, the *closed* attribute is set to $False$. if this parameter is used, it is also saved as species attribute of the Analyse class.
add_reaction: uses a given object of the reaction class to add to the list of reactions.
excluded_species: removes a given species from the support and production of all reactions. Mostly this was used to eliminate inconsistencies in reaction networks of SBML-files that specifically described a null species as support of inflow reactions and product of outflow reactions.
make_inflow_separable: can be set to True to allow for inflow to be just in at least one compartment, by changing the inflow reactions to a self replicating reaction. The algorithm guarantees for the reaction to be present in each SAR, by not changing the *reaction.always* attribute, as it is set to True. This allows for special separation of these reactions as shown in..

3.2.4 Function create_ERCs()

Input: list_Of_All_reactions
Result: dictionary of objects (class ERC) → ERC-dictionary
Parameters: none
The algorithm originates straight-forward from the definition of closure of the reactions. For each given reaction, a species set for its ERC is created by unifying its support and its products. Any supported reactions and their products are added successively in a recursive function. A counter remembers the last added reaction and starts from this counter, to avoid excessive checking of the reactions at the front of the list. The result is saved as an instance of an ERC class object. The ERC class contains the attributes *reactions* and *species*. All the ERCs are then saved in an ERC-dictionary, which is the means of the information transfer. The name of the starting reaction functions as key.
The ERCs are created within the function of the LP setup and is saved, since this object follows the definition of 4, but before being inserted as constraints, the transitive reduction of ERCs is applied. The old ERC_dictionary is still used for visualization and calculation of the average ERC length. Also this is the printable result, from the $print\_ERC()$ function of the *Analysis* class since its effect/impact is easier to read. The closure of reactions can include reaction cascades and inflow reactions. Different reactions can also produce similar or equal species spaces. All these motifs can result in overlap of constraints that are generated for the ERCs. The one-sided relation of ERCs follows the transitivity as shown here:

The ERC of a reaction R1 which includes a reaction R2, contains the species set of the reaction R2 and therefore contains at least all reactions included in the ERC of R2.
The algorithm of transitive reduction (Hsu 1975) is applied. The algorithm has a computational complexity of $r\hat{3}$ for a set of ERCs containing almost all reactions. To reduce computation time, the ERCs are converted into a boolean matrix of size r x r. It contains the boolean value if a reaction r2 is in the ERC of r1 in A[r1,r2]. After application of the

algorithm, we gain a transitive reduced list of ERCs, called reduced ERCs. This is then used for the implementation of constraints.

3.2.5 Solving for SARs (setup_LP(SAR))

Input: list of reactions
Result: list of SARs
parameters: only_largest_solution, second_optimize, see_constraints, apply_time_limit job (if ="DO") 3.2.7, use_Gurobi
To compute the SARs the function $set\_up\_LP()$ is used, which uses the ERCs and the reaction network by iteratively solving integer linear problems (ILPs). The LP is defined as a maximization problem. This results in the solution list being sorted by size, starting from the largest. The LP has 3 different objective functions, which map a search space onto a linear result vector $f(searchspace) -> max(goal)$:

objective function	search space	result	optimizer
max(SARs)	$v^r \cdot [0,1]^r$	[SAR][] for SAR/ $max(b(v^r))forr \in \mathcal{R}$ ∈solutions	CBC & Gurobi
maximize for SARs with max number of overproduction	$v^r \cdot [0,1]^n \cdot [0,1]^m$	SARs (reaction sets) with set of overproduction	CBC
maximize for DOs with max number of reactions	$v^r \cdot [0,1]^n \cdot [0,1]^m$	DOs (species sets)	CBC

The search space is defined by the list of variables, used for the LP. We differ between variables of the continuous and binary/boolean type. The linear constraints can be divided into 4 different sources:

constraint origin	max constraint number	only CBC	equation shape
stoichiometric matrix	$\|S\|$		$N \cdot v \geq 0$
ERCs	$\|R\|^2$		$b_j \geq b_i \quad \forall r_i \in R, \forall r_j \in ERC(r_i), r_i \neq r_j$
inflow reactions	$\|R\|$		$b_i = 1$ for all inflow reactions $r_i \in R$
integer cut of solution	$\|(max(20, \binom{R}{R/2}))\|$	x	$\sum_{r_j \in \mathrm{SAR}^i} b_j \leq \|\mathrm{SAR}^i\| - 1$ for all solutions $\mathrm{SAR}^i \in \{\mathrm{SAR}^1, \ldots, \mathrm{SAR}^k\}$
relation of boolean and real variable (used in 2 and 3)	$2 \cdot \|R\|+$ (optional: $2 \cdot \|S\|$)		$b_i \cdot c \geq v_i \geq b_i$ for all $r_i \in R$

These constraints are required to ensure the following properties:

- The stoichiometric matrix is translated to implement the self-maintenance property.
- The second constraints resemble the dependence of reactions following the ERCs.
- Inflow reactions have to be active, since they are also supported by the empty set, which is not included in other constraints.
- An integer cut of an already found solution to exclude it for the next solving. After 20 of these cuts, the next time a solution is found that has a smaller number of reactions or species (for SARs or DOs), the solver is build up again with now only 1 integer cut instead of the 20+. We do not need this in Gurobi because of the solution pool.
- The relation of the flux vector and its boolean variable has to be set. The boolean variables are used for anything besides the implementation of the stoichiometric matrix3.

Additional constraints can come from maximizing the species that are overproduced by a SAR by using the parameter second_optimize=True. This is done by changing the objective function to the second entry. To still keep the correct order of the SARs the number of overproduced species is devided by a sufficiently large constant which has to be at least greater than the number of species. As a simplification the constant 10000 is used. Additional constraints can also come from introducing a species subset and therefore inactivating violating reactions 3.2.3

The CBC solver outputs solutions, until it cannot find further solutions and returns the $infeasible$ status. It is to note that in a few cases the status $undefined$ appeared. This means, the LP can not guarantee to have found all solutions. But this has not occured with the current version of the algorithm.

3.2.6 Alternative Solver

Besides the use of the integrated base solver CBC_CMD of Pulp, you can use the Gurobi solver by using the parameter *use_Gurobi*=True. Even though the pulp interface allows to parse the problem directly to the solver, this approach was discarded, since this restricts some options of the Gurobi solver. Instead Pulp writes the Model as an *.lp* file and is read by the gurobipy read function.
This Solver differs in the processing of the solutions. This is mainly because of the use of the *Solutionpool* option of Gurobi. It allows us to extract not only the best solution, but all the solutions found while traversing to the optimal solution. By setting the $PoolSearchMode$ Parameter to 2, the LP guarantees that the extracted solutions are the most optimal solutions and there dis no solution with a higher objective value.
The use of this option does not use integer cuts for the reaction vectors but rather the whole objective function. This means when including the *second_optimize* parameter, we get a solution for each possible combination of overproduction. This could be circumvented by separating the two optimization processes, but the CBC solver should be sufficient to do that, when needed.

In the next subsection it is briefly described how the *set_up_LP*() function is modified to compute the DOs instead of the SARs.

3.2.7 Solving for DOs (setup_LP(DO))

Since now we solve for DOs, which are sets of species instead of reactions, we add boolean variables *s_exist*, which describes the existence of a species in a DO. These can be seen in the third entry of the objective function table. This is used to ensure the closure of the spaces these species create. To calculate the reactions of these spaces the function *create_ERCs*() is executed. The function starts with a single species instead of a reaction and the union of its support and products. A DO can be realized by several SARs, of which we are interested in the one with the most reactions since it is unique and contains all other SARs.

3.2.8 Function create_MRCs()

Following 5 the function *create_MRCs*() computes the (unique) maximal reactive compartments of a given SAR. We work on a list of candidates for the MRCs, which is initialized by $S(SAR)$.
The first property of the MRCs, to not support a reaction outside the SAR, is ensured by checking each candidate for the support of each not active reaction. A compartment which supports an inactive reaction of order k is split in k smaller compartments in the following way:

For a reaction r with support $supp(r)$ we separate a species set A the following way

$$A \xrightarrow{\text{splitt for r}} \{A \setminus \{s\} \mid s \in supp(r)\} \quad . \tag{27}$$

After that we eliminate candidates that are subsets of other valid candidates.
As the next step, the candidates are checked for closedness. Each reaction that extends the existing species set can not occur in this set and is therefore inactivated by updating the MRCs in the same way as done for the inactive reactions. In contrast to the processing of the inactive reactions, the property of being closed to an active reaction can change after the removing of species to achieve closedeness for another reaction. Therefore we need to loop over all reactions until a full loop is completed without changing a set. Finally, proper subsets of candidates are deleted again, if present.

3.2.9 Function get_minimal_compartments()

The newly created MRCs are now used in an ILP to gain the minimal number of compartments, needed for the affiliated SAR. The ILP, which is a set cover problem (Vazirani

2001), is as follows:

$$\min \sum_{MRC \in MRC\_list} b_{MRC} \tag{28}$$

$$\text{subject to} \sum_{MRC \in MRC\_list:\, species \in MRC} b_{MRC} \geq 1 \text{ for all } species(SAR) \tag{29}$$

$$\text{and to} \sum_{MRC \in MRC\_list:\, supp(reaction) \subseteq MRC} b_{MRC} \geq 1 \text{ for all } reaction \in SAR \tag{30}$$

We aim at finding a minimal set of MRCs, such that all species in $species(SAR)$ are covered and each reaction of the SAR can run in at least one MRC.

3.2.10 Function get_Lattice()

Input: list_of_DOs or list_of_SARs
Results: gv file (lattice as text) and pdf file (visualization of lattice)
The result can be changed by a multitude of boolean parameters to cover a range of needs.
$shortform$: names the reactions by indices instead of their original names.
$show\_species$: shows the species of SARs
$show\_new$:hides species and reactions already contained in subsets
$second_value$: shows overproduction of species in SAR-lattice (the $setup\_LP(SAR)$ function had to used with parameter $second\_optimize = True$) or largest SAR of the DO in the DO-lattice. use_naive: is used for data, that is not a lattice(see

There is also a 3 state parameter (can be True, number or False): $show\_compartments$: shows minimal compartments, just number of min compartments (set by default) or no information about compartments.

By solving for either the list of DOs or the list of SARs, we gain a set of solutions in a partial order. Moreover the set of DOs and the set of SARs are each always a lattice ((Stephan Peter, Bashar Ibrahim, and Peter Dittrich 2021) and 4). The list contains a smallest and a largest solution. Both list are completed for merge, but not for union. Also, they have an infimum for each element. By inverting each element of a list, we create a set that is complete for intersection and has a supremum for each element. With inverting an element, we mean set of reactions or species is mapped to a set of reaction of species, where the following applies:

$$x \in S; \tag{31}$$

$$S \setminus x = \bar{x} \tag{32}$$

With these attributes, we can apply the iPred algorithm of Baixeries et al. (2009). This algorithm achieves linear running time by saving the elements without subsets at each time step, as well as all supersets of each element.
As it turned out, a naive algorithm will result in roughly the same running time, for a small number of elements. "small" meaning that the diagram is visually comprehensible. But since parts of this algorithm are used for the analyze_DO() function, we can hope for a reduction of computation. The naive algorithm is attached as a function under the

name Create_hasse2() and can be used if one wants to create a diagram of an uncompleted SAR list. This algorithm builds up the lattice from smallest to largest vertex. Each vertex has to be checked for all smaller vertices. If the vertex is not a subset, it continues to the next. However, if it is a proper subset, it creates an edge with it and gains the coverage of the vertices that are a subset of the connected vertex.

3.2.11 Analysis Class

The different functions regarding different aspects around the reaction network analysis can be managed by a class, which governs/steers the variables. It saves any information in class attributes and pipes them into their advancing/ processing functions. This allows for a clear and simple communication with the user as well as the managing of several reaction networks simultaniously.

3.2.12 Function setup_LP_DOs_with_SAR()

3.3 Runtime Analyses

3.3.1 Termination

The functions *get_reactions*() and *change_network*() terminate since the set of species and the set of reactions are finite by definition.

This also guarantees that the function $create\_ERCs()$ terminates.

For the LP function, we have to differentiate between the Pulp and the Gurobi solver. At first, the Pulp interface $setup\_LP()$ loops over finite sets of species, reactions, and ERCs, that create a finite number of constraints. The maximal number of SARs is 2^R and the number of DOs is 2^S. These solutions can be found by the algorithm and take exponentially more time when using the integer cuts. In each iteration the number of remaining solutions is decreased by 1, until no more solutions can be found. It is to note, that in a few cases the status $undefined$ appeared after finding several, but not all solutions. A special output is emitted if that is the case, but with the latest version these problematic system terminated properly. If a model is not solvable, it is recommended to exclude the maximization for the second value or use the Gurobi solver. Using the solutionpool of Gurobi, we have again 2^R maximal solutions. But since we now enter a finite number of maximal solutions as parameter, the termination process is even clearer.

get_lattice() uses the finite number of solutions (SARs or DOs) to form the lattice. Since the number of the vertices of the lattice is finite and there cannot be more than one edge between two vertices, *get_lattice*() terminates.

The function $create\_MRCs()$ creates a number of sets of species that is limited by the number of inactive reactions and their finite support. These are then checked for

closedness and subset property. The finite set of MRCs as well as the finite sets of SAR and species(SAR) result in a finite number of constraints.

One solution with the minimal number of compartments is then returned by $get\_minimal\_compartments()$.

3.3.2 Time Complexity remarks

Given a reaction network $(\mathcal{S}, \mathcal{R})$, we want to remind the reader of the following variable names:

- $n = |\mathcal{S}|$ = number of species,
- $m = |\mathcal{R}|$ = number of reactions,
- $k = max\{max\{|supp(r)|, products(r)\} : \ r \in \mathcal{R}\}$ = highest order of a reaction (it follows: $k \leq n$),

The algorithm of $pipeline\_DO()$ often uses the python function $set\_1.issubset(set\_2)$, which we assume to have a time complexity of $O(|set\_1|)$.

3.3.3 Time Complexity of getReaction()

This function creates objects of the reaction class by iterating through support (k) and product (k) of every reaction (m). Therefore, its time complexity is

$$O(m \cdot (k + k)) \in O(m \cdot k) \tag{33}$$

The optional addition of separate reactions to include their reversibility iterates over all reactions and has to check its products and support against the support and products of every over reaction to rule out identical reactions:

$$O(m \cdot m \cdot k \cdot k \cdot k \cdot k) \in O(m^2 \cdot k^4) \tag{34}$$

3.3.4 Time Complexity of change_network()

Here we have to differentiate between the possible parameters, which crucially differ in their functions as well as in their time complexity.

parameter	workflow	runtime complexity
subset of species	iterates through all reactions (m) and checks if the support is subset of the given species set(k) and if the products (k) violate its closedness.	$O(m \cdot (k+k)) \in O(m \cdot k)$
add reaction	iterates through support (k) and product(k) of a reaction	$O(k+k)$
remove species	iterates through all reactions (m) and checks for occurence of the species (k+k), before removing it	$O(m \cdot k + k) \in O(m \cdot k)$
make inflow separable	iterates through all reaction and for reactions with an empty support, the species are added as reactants and the stoichiometric parameters of the product changed	$O(m \cdot k + k) \in O(m \cdot k)$

3.3.5 Time Complexity of create_ERCs()

For every reaction (m), a species set is created, then it checks for every other reaction (m), if its support is within the set (k). After adding a reaction, the process is repeated with the remaining reactions, up to a maximum of all reactions (m). Therefore we gain the following time complexity:

$$O(m \cdot m \cdot m \cdot k) \in O(m^3 \cdot k) \tag{35}$$

The transitivity check then iterates over all ERCs(m), checking for every contained reaction(max m), if it is in the ERC, of another reaction within the ERC(m). By implementing the data as a matrix, we gain:

$$O(m \cdot m \cdot m) \in O(m^3) \tag{36}$$

This is shown in (Hsu 1975) as well.

3.3.6 Time Complexity of setup_LP()

The usage of $create\_ERC()$ has a running time complexity of $O(m^3 \cdot k)$. The implementation of constraints is determined by the number of constraints and results in $O(m^2 + S)$.

The actual solving of the ILP has exponential time complexity since it is NP hard. (Papadimitriou and Steiglitz 1998). The complexity for a single solution is exponential with the number of boolean variables. This number differs for the different objective functions. When solving for SARs only, we have a variable for each reaction (m), while when including overproduction of species or solving for DOs, we also have variables for each species (n) resulting in

$$O(2^m) \text{ when solving for SARs and} \tag{37}$$
$$O(2^{m+n}) \text{ otherwise.} \tag{38}$$

When we solve for several solutions, we have no increase in complexity when using Gurobi through the use of the solution pool, while the usage of Pulp increases the complexity by solving up to 2^m (for SARs) or 2^n when solving for DOs.

Here we will have the largest difference between worst case runtime, and the application on real problems. As can be seen in (cf. 4.3), the actual solving time is impacted by the number of constraints as well as the number of boolean variables.

3.3.7 Time Complexity of get_lattice()

Given a sorted list of g solutions (of length $f \leq 2^m$) of SARs or DOs, we check the element to be the superset of every already placed vertex to link the vertices by an edge. This is done up to a number of $\frac{m \cdot m+1}{2} \in O(m^2)$ times. The actual check to be a superset is done in $O(m)$.
We use the ipred algorithm with the runtime:

$$|g| \times \omega(L) \times |m| \tag{39}$$

where $\omega(L)$ is the width of L (which is at most $\binom{m}{m/2}$) (Engel 1997)

resulting in

$$\frac{m!}{m/2!} \cdot 2^m \cdot m \tag{40}$$

we can round this up to

$$2^m \cdot 2^m \cdot m \tag{41}$$

resulting in $4^m \cdot m \in O(4^m \cdot m)$

The time complexity of the naive approach is examined below.
For each input, we check against each already processed input, resulting in

$$\sum_{i=1}^{|C|} i - 1 \text{ subset checks.} \tag{42}$$

Since the first element is checked for $|g| - 1$ times and the second is checked $|g| - 2$ times etc., we result in:

$$\sum_{i=1}^{|g|} |g_i| \cdot |g| - i \tag{43}$$

subset checks, where $|g_i|$ is the size of solution g_i. For the full set of solutions we can round that up to

$$\sum_{i=1}^{|g|} \frac{m}{2} \cdot \frac{|g|}{2} \tag{44}$$

$$= |g| \cdot \frac{m}{2} \cdot \frac{|g|}{2} \tag{45}$$

with g being 2^m, the result is

$$2^m \cdot m \cdot 2^m \in O(4^m \cdot m) \tag{46}$$

3.3.8 Time Complexity of create_MRCs()

Given a single SAR that is possible to be performed by a distribution of species $s_{SAR} \subseteqq \mathcal{S}$ with a set of active reactions $SAR \subseteqq \mathcal{R}$ and a set of inactive reactions $\hat{r}_{SAR} \subseteqq \mathcal{R}$ with

$$SAR \cup \hat{r}_{SAR} = \mathcal{R} \tag{47}$$

$$SAR \cap \hat{r}_{SAR} = \emptyset \tag{48}$$

The $S(SAR)$ is split up against the support of length k of all $\hat{r}_{SAR}$ as shown in 27. In the worst case the supports of all reactions are pairwise disjoint. If, for simplicity, they have the same length, we separate all species sets for the support of each reaction resulting in $k^{|\hat{r}_{SAR}|}$ subsets. But note that the number of these subsets is limited to 2^n. The number of splitting operations is given by

$$\sum_{i=1}^{|\hat{r}_{SAR}|} k^i. \tag{49}$$

In each step we create k copies. Then we remove in each copy one of the supporting species of the reaction. The time to copy an object is in O(n) resulting in a computation time of

$$\sum_{i=1}^{|\hat{r}_{SAR}|} k^i \cdot (n - i + 1). \tag{50}$$

After that, the set is checked for subsets. To show the dependence on more straightforward variables, we will instead use the upper bound of 2^n sets. These sets are compared with one another resulting in

$$(2^n)^2 \in O(4^n) \tag{51}$$

The next step is to check for closedness. The number of sets after deleting subsets has an upper bound of $\binom{n}{n/2}$. If the supports of the active reaction are pairwise disjoint, we create up to

$$\binom{n}{n/2} \cdot k^{|r_{SAR}|} \tag{52}$$

species sets. At each point of the closedness check, this set has an upper limit of 2^n sets. In contrast to the check for inactive reactions, we have to loop over all reactions, until none of them changes for an entire loop. This can take up to $|SAR|$ loops when only one reaction is changed for each loop. This results in

$$|SAR| \cdot |SAR| \cdot 2^n \cdot k \tag{53}$$

split operations each of which has time complexity$O(n - i + 1) \in O(n)$. This results in a total time complexity

$$O(|SAR| \cdot |SAR| \cdot 2^n \cdot k \cdot n). \tag{54}$$

This is $\in O(m^2 \cdot 2^n \cdot k \cdot n)$ At last, we have to add $O(4^n)$ for the final subset check. After the subset check we have an output of at most $\binom{n}{n/2}$ candidates.

3.3.9 Time Complexity of get_minimal_compartments()

Building up the $n \cdot m$ constraints of the ILP is of linear time complexity ($O(n+m)$). Solving the ILP is exponential, that is, $2^{\text{number of MRCs}}$ which is at most $2^{\binom{n}{n/2}} \in O(4^m)$. Since the problem resembles the set cover problem, which is proven NP-complete, it is not possible, to solve this problem in polynomial time.

4 Examples and Evaluation

This section showcases the appliance of the algorithms onto specific models. Trying to generalize the information of the list of SARs and the resulting lattice into specific quantitative attributes is challenging, because the dimensions of results differ for each network. Thus these quantitative attributes are broadly organized into three groups: properties of the network, properties of the SARs, and properties of the algorithm. This information helps to reference the more complex attributes.

Following the new approach of distributed organizations, we focus on SARs that can only occur through separation of species, called SAR_{DO}. Those SARs can therefore only originate from flux vectors that are viable in a pure DO, that is, a DO that is not an organization. Formally, a SAR is called a SAR_{DO}, if $R(S(SAR)) \neq SAR$, with $S(SAR)$ being the species of a SAR and $R(A) = \{r \in \mathcal{R} : supp(r_j) \subseteq A\}$ being the reactions that can happen with species A. Otherwise, a SAR is called an SAR_O, that is, when $R(S(SAR)) = SAR$.

Both these SARs can be given to the algorithm for calculating the minimal compartments. We call this attribute $mc(SAR)$. Through the definition of SAR_O it follows, that $mc(SAR_O) = 1$.

When interpreting the SAR as part of the lattice of SARs, we can also attain the number of new reactions, which do not exist in a subset SAR of the SAR (marked green in the Hasse diagram).

4.1 Interpreting the Lattice of SARs

At first we want to showcase an artificial model originating from the 2 generator model. We call it *IllustrativeExample*. This model is defined by its list of reactions, see below. The model consists of 3 generators, called A1, A2 and A3. These produce their corresponding products B1, B2 and B3 through the reactions R1, R2 and R3, respectively.
The generators are harmed by exposure to a foreign generator R4 or foreign products R5. At last reactions for the reproduction of the generators and the communication between the products of the generators is implemented. The largest solution is supposed to be a SAR_O to demonstrate the importance of underlying $SARs_{DO}$ which can be achieved by different levels of separation.

$r_1 : A1 \longrightarrow A1 + B1$ (generating function1)
$r_2 : A2 \longrightarrow A2 + B2$ (generating function2)
$r_3 : A3 \longrightarrow A3 + B3$ (generating function3)
$r_4 : A2 + B3 \longrightarrow B3$ (consuming interaction1)
$r_5 : B1 + A2 \longrightarrow \emptyset$ (consuming interaction2)
$r_6 : B1 + B3 \longrightarrow 2B2$ (consuming interaction3)
$r_7 : A1 + A2 \longrightarrow \emptyset$ (evaporation of products)
$r_8 : A1 + A2 + A3 \longrightarrow D1$ (production of valuable species)
$r_9 : D1 + B1 \longrightarrow 2B1 + A2$ (production of first generator)

After getting the solution for the SARs, the Hasse Diagramm is created with all optional parameters to show all available data. These include species, overproduction and a minimal number of MRCs to perform the SAR.

We see that all 3 generators can exist independently and build up the first level of reactive compartments. These can be combined for the first level of SARs that can be only obtained in a DO, as can be seen by the round shape of the knot as well as the number of compartments being larger than 1. An outflow reaction R4 can then be activated by introducing the second generator with the product of the first. Optional reactions like these can often inflate the number of SARs, but the support of this reaction is part of the support of the reactions that are required for larger SARs. This results in a thinner Hasse Diagram that is easier to interpret. Three SARs are only possible in three separated compartments. This is a rare occurrence if compared with the results of the BioModels database 4.2.9. This is because there has to be a specific pattern of reactions so that neither compartment can be merged with another one. Since these 3 possible merges ((a,b),(b,c) and (a,c)) have to be impossible, the minimum requirement for this to happen are 6 reactions of order two or four reactions, if reactions of order 3 are allowed. In the example both minimal requirements exist. Firstly, it is done by the combination of reactions R1, R2, R3, R4, R5 and R6 and the ladder is created by R1, R2 and R3, while R8 is supposed to be inactive.

The red lines indicate that two SARs have the same S(SAR). This is done to look for possible activation/deactivation of reactions in systems through a change in species distribution. The generation of the DO-lattice is possible as well, but since the reactions are mainly not triggered by a single species, the number of the DOs is 54. This is close to all 64 possible species sets (2^n) if non reactive DOs are included (38 if reactive DOs). Since the algorithm looks for the most reactive SARs for each of these DOs, the two $SARs_{DO}$ that are achieved by the two different levels of separation are not found. We can also see that the only species that trigger a reaction are the 3 generators. This fact becomes apparent, since these species can not be added without changing the associated SAR to contain there respective generator. It is to note that this information can only be read through the DO lattice. This reaction network is also the default network for if no SBML file is given. So all the results can be recreated.

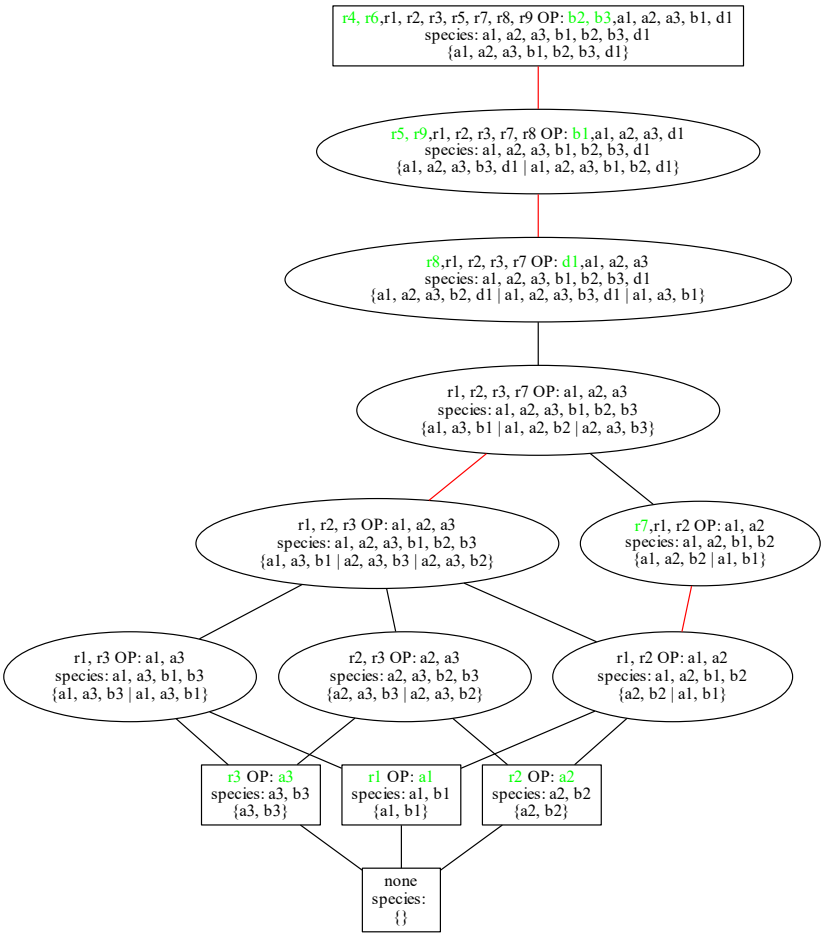


Figure 5: The Hasse Diagram of the Illustrative Example with 13 SARs and 3 sets of SARs with the same S(SAR) indicated by the red lines. For each vertex we see the SAR, there unique set of overproduction, the species set S(SAR) as well as a valid compartmentalization using the MRCs.

4.2 Bio Models Database

4.2.1 Approach

All data is generated by the function *iterate_over_database*, which takes a folder of SBML-files and uses the analysis class and all their functions to put together general information about the network, the ERCs, the SARs as well as their compartmentalization. Trying to generalize the information of the algorithms into specific countable attributes was a challenging aspect, since the dimensions of results differ for each network. Here is a broad summary of the properties explained in 4.1.

SAR property	description
SAR_{DO}	SAR that can only occur through separation of species
SAR_O	SAR that is not a SAR_{DO}
nR(SAR) $\in \mathcal{N}$	number of new reactions that do not exist in a subset SAR of the SAR (marked green in Hasse)
$S(SAR)$	minimal species set that can perform the SAR
$mc(SAR)$	minimal number of compartments of SAR

These attributes are used to count specific events in the model that are deemed meaningful in regard to new behavior through the integration of distribution of species. The result was this header of the exel table, to accommodate it to each of the 1052 models (state of 2/2023).

attribute	description
ID	ID of the sbml
Name	name of the model and suffixes for model changes (change inflow, different reverse)
S	number of species
R	number of reactions
inflow	number of inflow reactions
reverse	number of reversible reactions
#_SAR_O	Anzahl aller SARs die SAR_O sind
#_SAR_DO_only	Anzahl aller SARs die SAR_{DO} sind
new_rea_SAR_O	$\sum_i^{\lvert SAR_O \rvert} bool(nR(SAR_i) > 0)$
new_rea_SAR_DO	$\sum_i^{\lvert SAR_O \rvert} bool(nR(SAR1) > 0)$
s(DO)=s(O)	$\sum bool(s(SAR1) = s(SAR2)); \; for SAR1 \in SAR_{DO}, SAR2 \in SAR_O$
s(DO)=s(DO)	$\sum_{SAR1 \in SAR_{DO}} \sum_{SAR2 \in SAR_{DO}} bool(s(SAR1) = s(SAR2))$ for SAR1!=SAR2
largest_SAR_is	$bool(max(SAR) \in SAR_{DO})$
SAR_Os	SAR_O
SAR_DOs	SAR_{DO}
DOs	number of species sets that perform a SAR(including nonreactive)
comp3	$mc(SAR_i) = 3 for SAR_i \in SAR_{DO}$
comp4	$mc(SAR_i) = 4 for SAR_i \in SAR_{DO}$
comp5+	$mc(SAR_i) > 4 for SAR_i \in SAR_{DO}$
Timer_ERC	time of ERC calculation in second
avg_ERC_lenght	Average size of an ERC divided by the number of reactions of the whole network. Measures the fraction of reactions of the network covered by an ERC on average.
#constr	number of constraints of LP problem before the first solve
Timer_LP	time of solver in seconds
timer_MRC	time of *get_min_comp*() function for all SARs in seconds
timer_all	time of processing reaction network

The entries of new_rea_SAR_O, new_rea_SAR_DO, s(DO)=s(O), s(DO)=s(DO), largest_SAR_is are investigated in the analyze_SAR_set() function. After generating the list of SARs, it uses the *create_hasse*() function to mark reactions in a SAR that did not occur in its subsets. The entries of s(DO)=s(O),s(DO)=s(DO) have the least intuitive relevance for the behavior of a set of SARs, but addition and culling of species between compartments at critical molecular switch states could have significance for a model.

As we will discuss the handling of inflow reactions in 4.2.3 and reverse reactions in 4.2.4, there are 2 options to handle these respectively. To clarify the information that each of these versions yield, there is a separate evaluation of the model for each combination of the options. The options of a line in the exel table can be read through the appendix of the name, as the alternate reversible reaction approach will append "alt_rev" and the changed inflow will append "inflow" to the name, resulting in up to 4 iterations of one model.

Similar debatable topics are the handling of constant species as well as the species "EmptySet" in reactions. In the end this idea was discarded.

4.2.2 Correctness of Os

An additional part of this paper is the appliance of the SARs and their yield of information to already introduced models. We decided to take models of the BioModels Database, since a similar approach was already done by Kaleta, Richter, and Peter Dittrich (2009) to study the appearance of Os. This allowed to compare the results to verify the correctness for at least all SARs of Os of the 172 models that were available at the time. Since 2009 all of these models have been updated at some point. While the modified models will still have similar dynamics, small changes of reactions can lead to large deviations in the number of Os. By using the getModifiedDate() function of the libsbml package, it turned out, that all networks have been modified at some point since 2009. For that reason a second analysis is done with the older SBMl files, which are still available in the history section of the models. Kaleta made several alterations to the reaction network ,these include:

- the addition of an inflow and outflow reaction for each species that is marked as *constant* in the SBML file
- the addition of a second reaction for each reaction, that is marked as *reversible* in the SBML file, by switching the reactants and products
- changes to specific reactions that have dynamics on the level of interacting species hidden in the reaction kinetics

The last factor could not be taken into account, but Kaleta has data for the number of Os before and after this change. Since the old data of Kaleta was found on the server of the bioinformatics chair, we could also examine the original sbml files. To ensure that our algorithm has the same prerequisites as the algorithm of Kaleta, we first wanted to check if the *get_reaction* function extracts the same number of species and reactions as noted in the paper of Kaleta. The data only matched for about 70% of the models. What followed was a number of attempts while changing the interpretation of species, that are initialized

with a concentration greater than 0, as well as circumstances, in which the inflow and outflow reactions are not added. The results concluded, that the original files differed from the files extracted from the BioModels database with the original files resulting in more similarities with the given data of Kaleta in the dimensions of number of reactions and number of species. To classify the differences of the number of reactions and number of species, a matrix is created that plots the relation when using the comparison operators.

The results concluded, that the original files differed from the files extracted from the BioModels database with the original files resulting in more similarities with the given data of Kaleta in the dimensions of number of reactions and number of species. Although the folder only provided the data until model $BIOMD0000000150$, it has the highest percentage of correct models. The checks of the organizations are then made for just those models. The folder of the original data also contains files, which contain the Os of the models. These Os are named by their names instead of IDs, which are usually used in the reaction of SBML.

These files were created by adding the reversible reactions, but skipping the addition of inflow and outflow reactions for constant species as well as species which have an initial concentration greater than 0. When the mentioned reactions were added, about 25% of the models were corrupted and had the wrong number of reactions, while only 2-4 new models attained the correct number of reactions. This suggests, that the handling of the reactions by Kaleta was not perceived correctly and the models that do not attain the correct numbers probably also fail to imitate the behavior of the reaction networks used by Kaleta.

Kaleta original Data #135

	K<R	K=R	K>R
$K < S$	0	0	0
$K = S$	6	84	41
$K > S$	2	0	2
		62%	

old sbml Data #167

	K<R	K=R	K>R
$K < S$	11	10	3
$K = S$	11	76	49
$K > S$	2	1	4
		45%	

current sbml Data #171

	K<R	K=R	K>R
$K < S$	1	1	3
$K = S$	8	99	57
$K > S$	0	2	0
		56%	

From these 135 models, 7 did not terminate properly, since the number of Os or DOs is of combinatoric complexity and terminated, since their number of SARs exceeded 4000. From these models, one belonged to the 84 models with the correct number of reactions and species.
The 83 models matched in 82 cases with the solutions given by Kaleta. The one exception was the data of Biomodel $BIOMD0000000041$. It had the same number of Organizations, but they differed in their set of species. This was because of a different handling of a species, that was only defined in reactions.
So, the algorithm is able to compute the 260 Os. This is a strong indication for the solutions to be correct as well as its significance as alternative algorithm for calculating all Os of a system.

To showcase the changes made in the SBMLs, we use the algorithm for the current files.

4.2.3 Handling Inflow Reactions

An inflow reaction like $\rightarrow s$ usually implies that species a is always present. However, when considering a compartmentalized system, we can imagine that the inflow can also be localised and thus constraint to particular compartments. This section shows how to handle a localised inflow and how this localised inflow can give rise to "novel" distributed organizations.

In order to localize the inflow, we simply replace each inflow reaction of the form $\rightarrow a$ by a self replication reaction of the form $a \rightarrow 2a$ and assure that $a \rightarrow 2a$ is always present in any SAR. This is achieved in the tool, by setting $make\_inflow\_separable = true$ (cf. 3.2.3).

With the localized inflow we change 549 models. After the changes 287 had a $SARDO$. Overall the models contained 114582 $SARs_{DO}$

4.2.4 Handling reversible Reactions

The most straight forward implementation of reversible reactions is the splitting into two seperate reactions as can be seen in Kaleta for example. It is used ensure stable fluxes without reducing the capacity of the model to portray all possible dynamics. The extension on the level of the reaction network also does not create inconsistencies within the processing. But its applicability to the dimensions of DOs or rather SARs bears the problem of inflation of viable solutions. This is showcased for the model BIOMD0000000092, with the reaction network of:

$$
\begin{aligned}
&v_1 : z \longleftrightarrow w + e && \text{(generating function1)}\\
&v_2 : z + e \longleftrightarrow ez && \text{(generating function2)}\\
&v_3 : ez \longleftrightarrow w + 2e && \text{(generating function3)}
\end{aligned}
$$

When using the normal definition of reverse reversible we gain the following two SARs: v1, v1_reverse, v2, v2_reverse, v3, v3_reverse, Neither of the two show the possible two maintaining cycles (v1_reverse,v2,v3,v1, v2_reverse, v3_reverse. This can be emphasized, by allowing for these reactions to be separable. We do so by adding a reaction specific enzyme to the side of reactants and products of each reaction. This results in the following diagram 6.

This demonstrates, that the found organization is the closure of the 3 reversible reactions. This means when including reversible reactions, while maximizing for reactiveness, we allow for these reactions to act as independent organization, which can coexist in every DO that does not contain all reactions of it.

That contradicts the initial idea of the SARs to only show viable behavior, while reducing trivial information. By only allowing for these reactions, to be performed in one direction, it is prohibited that the two active reactions cancel out each others species conversion. This restriction does not contradict the restraints of closure that the ERCs

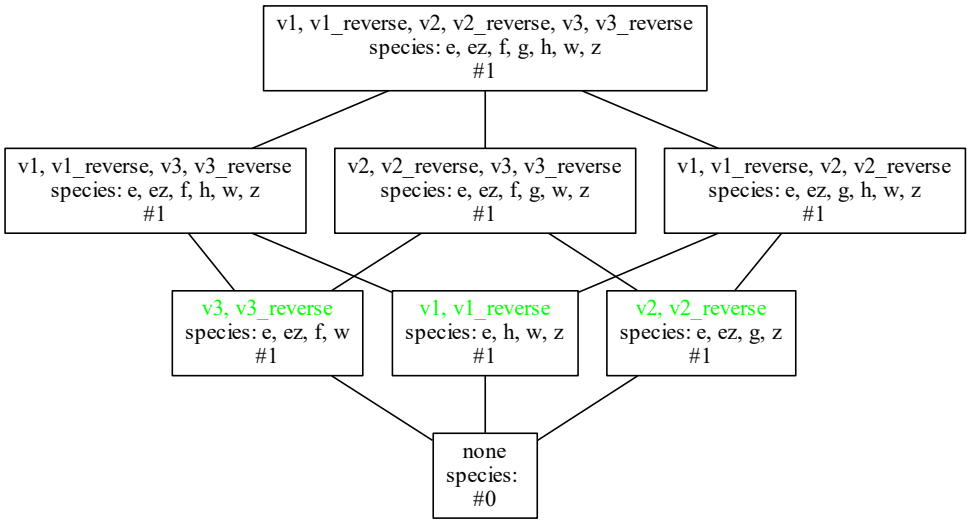


Figure 6: The Hasse diagram of a model with 2 directional loop.

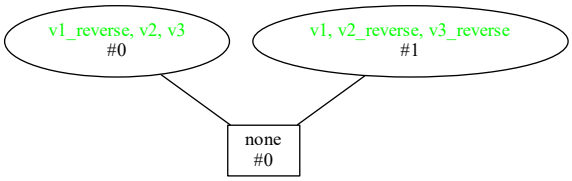


Figure 7: Hasse diagram of 3 reversible reactions with different interpretation by allowing one directional reaction flow.

should enforce, since the flux of the vector can be seen as the difference between the fluxes in an actual compartment. However, it contradicts the independence between the distributed compartments, as a reversible reaction, that is active in a compartment, should be able to have its reversed reaction active in a different compartment. This can be seen in the result below.

Note, that the normal function to create the Hasse diagram is not valid anymore, since the SARs do not resemble a lattice anymore. To circumvent that, the naive algorithm is used.

The algorithm has to apply a number of changes to translate this behavior. These are the following: at the step of reading the sbml document, instead of creating the reactions with the opposite directions, the reaction is marked with its attribute *reversible*. The reactions are created after the ERC are calculated and the implementation of ERCs is changed in the LP by requesting the sum of the boolean values of both functions to be included. Also, an extra constraint is implemented, which limits the sum of the boolean vectors to be at most 1.

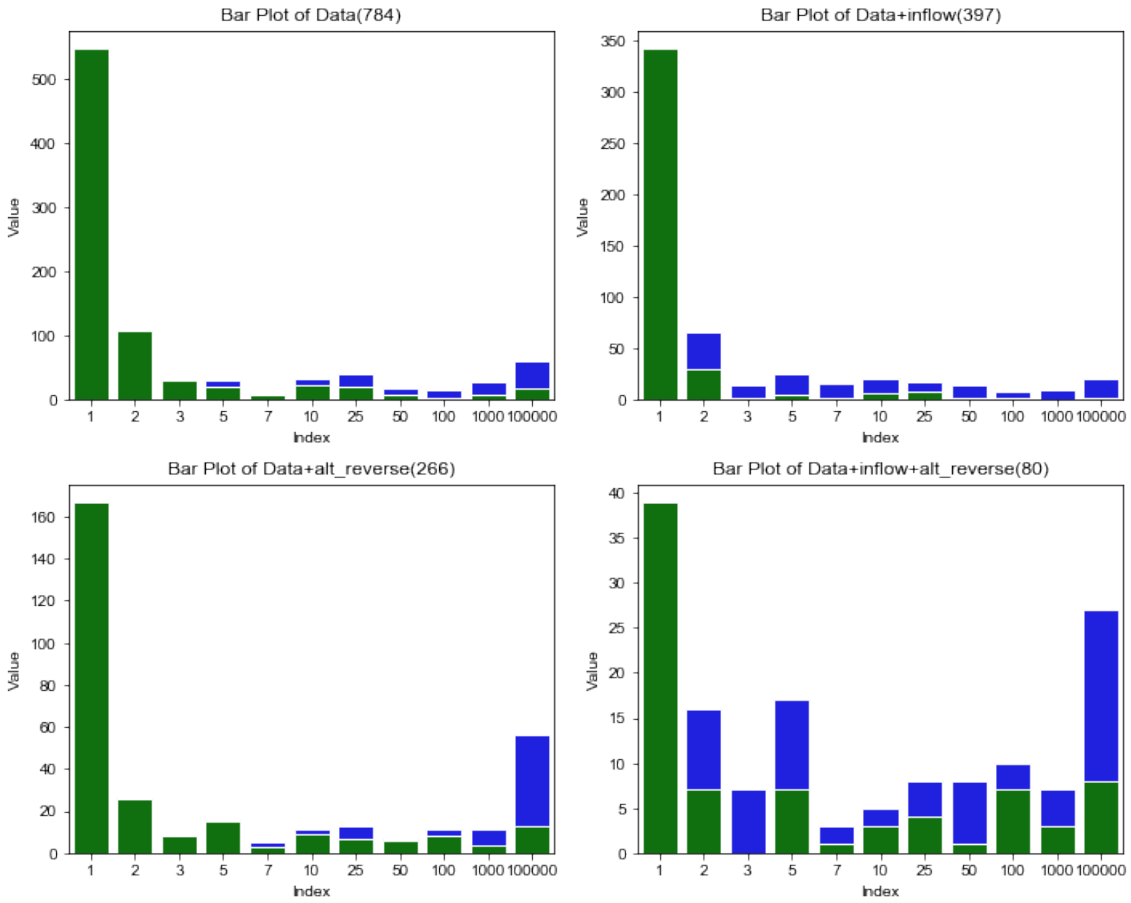


Figure 8: Plots of runs with different modifications to the reaction network. The models are counted for their number of SARs. The modifications are shown on top of the plot. Models that have at least one SAR_{DO} are marked blue.

In summary, the more straight forward approach when iterating over the BioModels database, but whenever we encounter large numbers of SARs that could be the result of combinatorics of reactions, the second implementation is applied to analyze the effect.

4.2.5 Results of the Database

In total the algorithm was able to calculate 467493 SARs, of which 198437 belong to a DO and 269056 can be fulfilled in an O. 130 Models were skipped since they had reached the 3000 SAR limit. Given that these Models have large numbers of reactions, we can assume, that the full numbers of SARs of these models could be in the multitudes. In the unchanged models, 59243 SAR_O and 114582 SAR_{DO} have been calculated.

Also the models should not have more than 50 SARs since the analysis would take too much time and its more likely, that these high numbers are achieved through combination of smaller SARs.

4.2.6 Further Examination

A total of 27 models were then chosen to be examined further by creating their respective Hasse Diagram. By considering their ERCs, reaction network as well as the paper that refer to the model, we hoped to gain new insights to several models. We will also try to derive attributes of the model, by identifying specific patterns seen in the Hasse Diagram, like spatial and temporal oscillation. It was then to filter out SARs that are not of interest. These included: SARs that only add reversible processes in a separate compartment without exposing them to their intended reaction partners. Also, flat or thin Hasse Diagramms can be left out, since they lack meaningful connections of the states. The former represent only shatters of a full O, whereas the latter cut reaction cascades by compartmentalization until the full O is build up.

The SBML models of the following IDs were examined manually: 7, 21, 26, 35, 41, 48, 92, 139, 147, 151, 163, 169, 186, 241, 327, 430, 431, 439, 489, 563, 599, 612, 630, 695, 862, 966 and 1002. None of them yield unique information of biological significance in their $SARsDO$. The quadruple iteration made it hard to interpret the given data, since they alter the landscape of SARs independently. The separation of inflow was able to lift constraints from the model and could have significance, when a species of an inflow reaction would have in inhibiting cascade of reactions. However, there were a significant number of models that showed SARs of a DO with new reactions that were the result of separating different inflows to inhibit basic reactions of the network. If these basic reactions had different supports of inflow, the lattice showed combinations of these partially separated basic reactions, until the first organization was reached. For example, this can be seen in model 42 by Nielsen et al. (1998) or is taken to the extreme in the models BIOMD0000000056.xml with 496 SARs of DOs.

In contrast to the expectation of the different handling of reverse reactions, we also encountered combinatorial numbers of SARs with the alternative handling. These happened in models that had a species closure which triggered a number of reactions, that could arrange self-maintenance through several smaller self-maintaining units of reactions. If all reactions of this units are reversible, each orientation of these units can be combined with one another. This can be seen in the model BIOMD0000000001.xml of Edelstein et al. (1996) or BIOMD0000000032.xml and BIOMD0000000051.xml.

Two different patterns, which occurred multiple times, were the addition of appendable sets of reactions to a basic SAR. These reaction sets were then independant from one another, resulting in a wide lattice with a polynomial number of SARs or one-sided dependant, resulting in a stretched lattice. These phenomenons can be seen for example in the model 139 of Hoffmann et al. (2002) or model 186 by B. Ibrahim et al. (2008).The latter can also be seen in model 326 or model 42. Some of the models that showed a number of SARs exhibit oscillation (5,35) or contained bistability (26). But the given SARs did not have any information regarding that aspect. The occurence of these models is rather due to a high ammount of models with these dynamics in the BioModels database or the build up of a model with multiple stable points has a positive effect on the ocurrence of SARs.

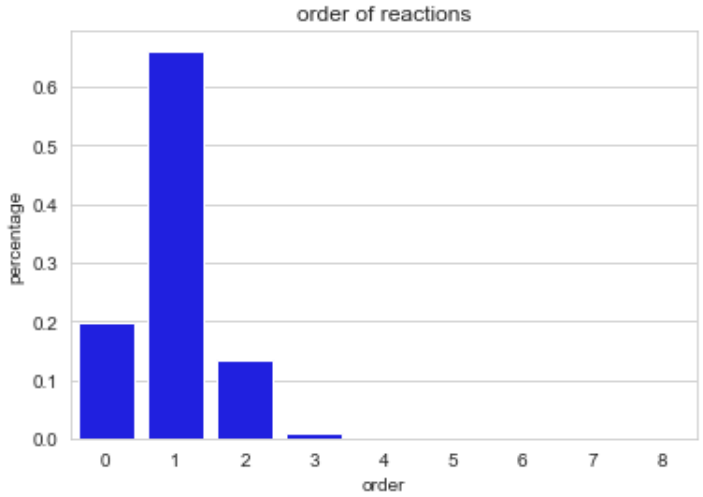


Figure 9: Distribution of order of reactions for all models of the Biomodels Database. With the reaction of order 1 dominating the models of the BioModels database.

4.2.7 Separation of Support

The lack of newly attained SARs can also come from the lack of separability of reactions. Since SAR_O arise from separation of supports of reactions. We will therefore look at the order of reactions and their influence on separability or rather their influence on the ERCs.
supp(r)=0: does not allow for inactivation in any compartment. The reaction is part of every ERC. Therefore it drastically increases the avg_ERC_length.
supp(r)=1: does not allow for inactivation of a reaction in a compartment of a reaction that has this species in list of reactants or products. This means that the reaction can also be excluded in an O, if the species is not contained in this species set
supp(r)>1: the species can be separated in several compartments to be deactivated. We therefore might attain a SAR_{DO}. To reduce the impact of large reaction networks, we look at the average distribution of all models. 9 shows, that the number of reactions of order 0 and 1 outweigh the reaction of a higher order. This results in only 14 % of all reactions to be inactivatable in a SARDO exclusively. This does not even account for reactions which are inseparable through their dependence on inflowing species. It is to assume that the large amount of reactions of a low order is one of the main drivers for the low number of models, where SAR_{DO} occur.

4.2.8 MRCs

The rather large upper border of the algorithms to compute MRCs, as well as the possibly high number of MRCs, which serve as input for the LP problem, arise the question, whether or not the calculation time limits the applicability of the algorithm. And indeed, we see a timeout for the computation of the minimal compartments in several models. We even see that the number of MRCs can not be calculated for the models 469-473 of Smallbone and Mendes (2013), as the algorithm is not able to check for the subset property between the candidates. But besides that, we want to grasp the behavior of the MRCs for a model.

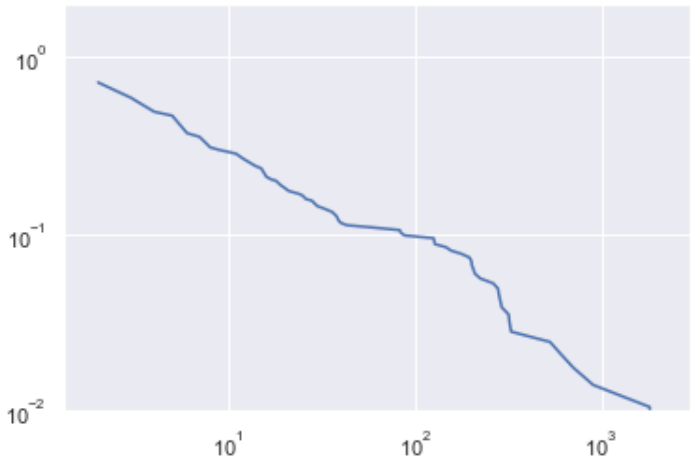


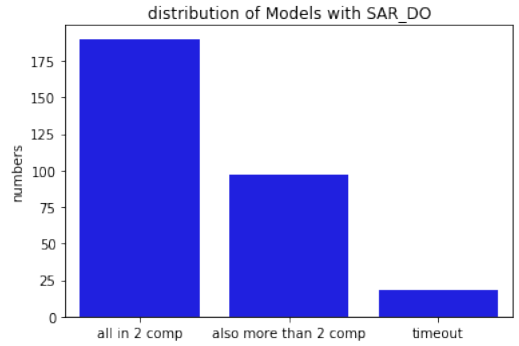
Figure 10: Distribution of the maximum number of MRCs, before computing the minimal set cover. It is shown as a linear function in a log-log graph, which suggests a power law distribution.

The computation of the compartmentalization is separate for each SAR. By marking the largest number of MRCs, we get the SAR with the highest number of boolean variables for the ILP of the minimal number of compartments. The number of MRCs correlates with the number of reactions of the model as well as the number of possible SARs. In Figure **??** we see that the maximal number of the MRCs of all models resembles a power-law probability distribution. To understand the distribution of MRCs within a model, two models that did not reach the maximum of SARs (3000) are examined. The following models have been chosen for the more detailed analysis. One of the largest MRCs of a model that had less than 3000 SARs, was model 667 with the max_mrc of 182. Since this model did not have SARs with the minimal number of compartments greater than two, model 399 is also included for this analysis, where 112 out of 806 SAR_{DO} have a minimal number of compartments of 3. The maximal number of MRC is 28.

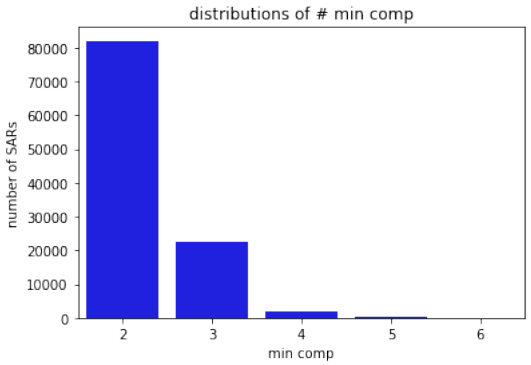
We see that the number of MRCs has the highest peaks around the Median of the SARs. This makes sense, since for SARs containing most of the possible reactions, the species set can not be split as much for all inactive reactions. On the other hand, SARs with only a few reactions can not achieve high numbers of ERCs, since the compartments contain far less species. This limits the the number of possible combinations of these species.

4.2.9 Compartmentalizations

In general, the algorithm determined 106764 specific compartmentalizations of SARs. These SARs had the following distribution:11b. The models could be divided into the 3 groups of containing only SARs with $mc(SARs_{DO}) <= 2$, containing at least one SAR with $mc(SARs_{DO}) > 2$ and models where the calculation of all SARs resulted in a timeout that was set to 180 seconds. 11a The largest portion of models does not contain compartmentalizations of more than 2. This is a hint for these models to only need to exclude a small number of reactions from one another. Against the expectations several



(a) The distribution of models according to their behavior according to the algorithm of computation of the MRCs.



(b) The distribution of the SARs according to their number of minimal compartments.

Figure 11: Distributions of models from BioModels with respect to their MRCs.

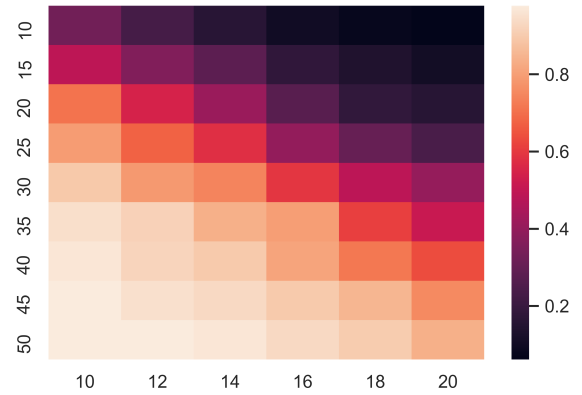
models had SARs with the minimal number of compartments going as high as 6 (one model) This is because with a higher number of compartments, the number of specific reactions to enforce this behavior rises exponentially. Since all compartments have to be unable to be merged with one another.

4.3 Randomly Generated Networks

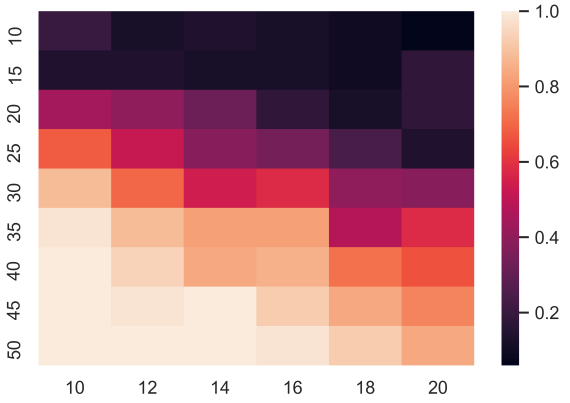
To verify some general insights gained from the iteration over the SBMML database, a random network generator made available by the group of Thomas Veloz (Vrije Universiteit Brussel) is used. This showcases the occurence of $SARs_{DO}$ in random models as well as the effect of specific parameters to gain complex DO behavior. As expected, Figure 12a shows that the average ERC length increases with a rising number of reactions and a falling number of species. Many models of the BioModels Database have a high average ERC length. Since these reactions are mostly mass preserving, we can get the empty SAR as well as a SAR of all reactions Figure 12b. because very often the highly interconnected species can produce one another sufficiently. When we look for models with more than 2 SARs, we see, that correlate with an ERC length between 0.6 and 0.7 Figure 12c. This is an indicator that random Models have a beneficial response, when there species are as connected as possible, while not being to restricted in their possible flux vectors. Therefore The avg ERC length is a measure of connectivity for within a network.

5 Conclusion

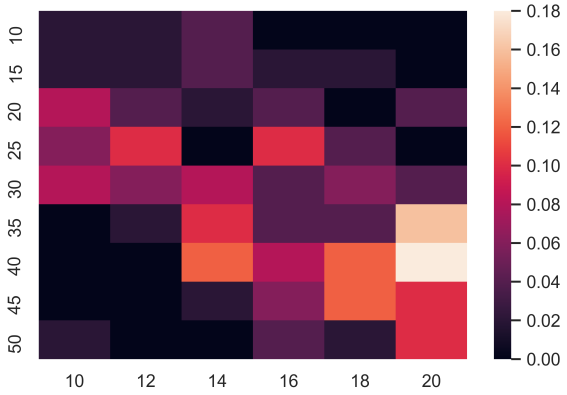
This thesis tackles the problem of distributed organizations from an algorithmic stand point. While the point of view of species sets was approached in a feasible manner, the new view from the sets of active reactions is placed next to it. The two new concepts of *elementary reaction closure* (ERCs) and *sets of active reactions* (SARs) are defined and mathematically formulated.



(a) Fraction of reactions of the network covered by an ERC on average (avg_ERC_length). It has the highest, when the highest number of reactions is distributed between the lowest number of species.



(b) The plot of the probability for a model, to have more than one SAR. It has the highest probability, when the ERC-length is high.



(c) The plot of the probability for a model, to have more than two SARs.

Figure 12: ERC size and SAR distribution in random models with different number of reactions (vertical axis) and number of species (horizontal axis). For each parameter combination 50 random models have been generated with the Velozgroup network generator.

The problem of compartmentalization has been disconnected to allow for an individual consideration and is solved with the help of an additional ILP. Introducing the *maximal reactive compartments* (MRCs) allows to analyze unambiguous properties of potential compartmentalization.

Besides these theoretical concepts, a practically oriented set of algorithms has been developed and is publicly available as a Python package to serve as a tool to infer structure from any reaction network. The package uses a user friendly managing class and a tutorial for an intuitive usage. The models can either be analyzed in a broad spectrum to scan through large systems, or in detail to gain a graphical informative representation in regard to distributed self-maintenance.

The Python code has been structured to represent the workflow of a typical analysis. A termination and complexity analysis provides insight to borders of applicability and proof correctness. Although the complexity of SAR computation and MRC computation turned out to be non-polynomial, the algorithm could be applied in practice, for example, 90% of the models of BioModels can be computed in seconds. The usage of two different linear programming solvers, allows to use an easy to install program, while allowing to increase the capacity for users, who want to deal with the subject in more detail. The gap in performance is greater than expected with Gurobi being about 1000 times faster and being able to solve ten times larger networks in the study presented here.

The application of the algorithms on the BioModels database is put into context of earlier scientific contributions. 60 % of the models posses only one organization (O) and one SAR. 15% of the models have at least one SAR_{DO}, that is, a set of active reactions that can only exist when distributing the species among at least two compartments. When considering a localized inflow, that is, an inflow localized to particular compartments, the number of models with at least one SAR_{DO} increases significantly, e.g., 50 % of the models with an inflow (60% of all models) posses an SAR_{DO} in that case. There are several sources for a combinatorial number of SARs, with reversible reactions being one of the most dominant factors, for example, model BIOMD0000000103.

In summary, this thesis has introduced novel theoretical concepts for providing a novel perspective on the structure of reaction networks with respect to compartmentalized dynamics, has implemented tools that show that these properties can be computed for the majority of models of BioModels database and that these models differ in these structural properties. The biological meaning of these structures remains open for future investigations.

6 Bibliography

References

Aris, Rutherford (Jan. 1965). "Prolegomena to the rational analysis of systems of chemical reactions". In: *Archive for Rational Mechanics and Analysis* 19.2, pp. 81–99. ISSN: 1432-0673. DOI: 10.1007/BF00282276. URL: https://doi.org/10.1007/BF00282276.

Baixeries, Jaume et al. (May 2009). "Yet a Faster Algorithm for Building the Hasse Diagram of a Concept Lattice". In: pp. 162–177. ISBN: 978-3-642-01814-5. DOI: 10.1007/978-3-642-01815-2_13.

Cardelli, Luca (Aug. 2014). "Morphisms of reaction networks that couple structure to function". In: *BMC Systems Biology* 8.1, p. 84. ISSN: 1752-0509. DOI: 10.1186/1752-0509-8-84. URL: https://doi.org/10.1186/1752-0509-8-84.

Centler, Florian, Christoph Kaleta, Pietro Speroni di Fenizio, et al. (2008). "Computing chemical organizations in biological networks". In: *Bioinformatics* 24.14, pp. 1611–1618.

Centler, Florian, Christoph Kaleta, Pietro Speroni di Fenizio, et al. (May 2010). "A parallel algorithm to compute chemical organizations in biological networks". In: *Bioinformatics* 26.14, pp. 1788–1789. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq263. eprint: https://academic.oup.com/bioinformatics/article-pdf/26/14/1788/48851837/bioinformatics_26_14_1788.pdf. URL: https://doi.org/10.1093/bioinformatics/btq263.

Dickenstein, Alicia et al. (May 2019). "Multistationarity in Structured Reaction Networks". In: *Bulletin of Mathematical Biology* 81.5, pp. 1527–1581. ISSN: 1522-9602. DOI: 10.1007/s11538-019-00572-6. URL: https://doi.org/10.1007/s11538-019-00572-6.

Dittrich, Peter and Pietro Speroni Di Fenizio (2007). "Chemical organisation theory". In: *Bulletin of mathematical biology* 69.4, pp. 1199–1231.

Dittrich, Peter and Lars Winter (July 2005). "Reaction Networks as a Formal Mechanism to Explain Social Phenomena". In: *H. Deguchi, K. Kijima, T. Terano, H. Kita (Eds.), Proceeding of The Fourth International Workshop on Agent-based Approaches in Economics and Social Complex Systems (AESCS 2005)*, pp. 433–446.

— (2008). "Chemical organizations in a toy model of the political system". In: *Advances in Complex Systems* 11.04, pp. 609–627.

Edelstein, Stuart J et al. (1996). "A kinetic mechanism for nicotinic acetylcholine receptors based on multiple allosteric transitions". In: *Biological cybernetics* 75, pp. 361–379.

Engel, Konrad (1997). *Sperner theory*. 65. Cambridge University Press.

Feinberg, Martin (Jan. 1972). "Complex balancing in general kinetic systems". In: *Archive for Rational Mechanics and Analysis* 49.3, pp. 187–194. ISSN: 1432-0673. DOI: 10.1007/BF00255665. URL: https://doi.org/10.1007/BF00255665.

— (1987). "Chemical reaction network structure and the stability of complex isothermal reactors—I. The deficiency zero and deficiency one theorems". In: *Chemical Engineering Science* 42.10, pp. 2229–2268. ISSN: 0009-2509. DOI: https://doi.org/10.1016/0009-2509(87)80099-4. URL: https://www.sciencedirect.com/science/article/pii/0009250987800994.

Figueiredo, Luis F. de et al. (Sept. 2008). "Can sugars be produced from fatty acids? A test case for pathway analysis tools". In: *Bioinformatics* 24.22, pp. 2615–2621. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btn500. eprint: https://academic.oup.com/bioinformatics/article-pdf/24/22/2615/49055538/bioinformatics_24_22_2615.pdf. URL: https://doi.org/10.1093/bioinformatics/btn500.

Fontana, Walter and Leo W Buss (1994). "The arrival of the fittest: Toward a theory of biological organization". In: *Bulletin of Mathematical Biology* 56.1, pp. 1–64.

Gatermann, Karin (July 2001). "Counting Stable Solutions of Sparse Polynomial Systems in Chemistry". In: DOI: 10.1090/conm/286/04754.

Gatermann, Karin, Markus Eiswirth, and Anke Sensse (2005). "Toric ideals and graph theory to analyze Hopf bifurcations in mass action systems". In: *Journal of Symbolic Computation* 40.6, pp. 1361–1382.

Hoffmann, Alexander et al. (2002). "The IκB-NF-κB signaling module: temporal control and selective gene activation". In: *science* 298.5596, pp. 1241–1245.

Horn, F. and R. Jackson (Jan. 1972). "General mass action kinetics". In: *Archive for Rational Mechanics and Analysis* 47.2, pp. 81–116. ISSN: 1432-0673. DOI: 10.1007/BF00251225. URL: https://doi.org/10.1007/BF00251225.

Hsu, Harry T. (Jan. 1975). "An Algorithm for Finding a Minimal Equivalent Graph of a Digraph". In: *J. ACM* 22.1, pp. 11–16. ISSN: 0004-5411. DOI: 10.1145/321864.321866. URL: https://doi.org/10.1145/321864.321866.

Ibrahim, B. et al. (2008). "In-silico modeling of the Mitotic Spindle Assembly Checkpoint". In: *PLoS ONE* 3.2.

Jiang, Chunheng et al. (2021). "Nuclear reaction network unveils novel reaction patterns based on stellar energies". In: *New Journal of Physics* 23.8, p. 083035.

Kaleta, Christoph, Stephan Richter, and Peter Dittrich (2009). "Using chemical organization theory for model checking". In: *Bioinformatics* 25.15, pp. 1915–1922.

Nielsen, K et al. (1998). "Sustained oscillations in glycolysis: an experimental and theoretical study of chaotic and complex periodic behavior and of quenching of simple oscillations". In: *Biophysical Chemistry* 72.1-2, pp. 49–62.

Papadimitriou, Christos H and Kenneth Steiglitz (1998). *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

Peter, S. and P. Dittrich (2011). "On the Relation between Organizations and Limit Sets in Chemical Reaction Systems". In: *Advances in Complex Systems* 14.1, pp. 77–96.

Peter, Stephan, Peter Dittrich, and Bashar Ibrahim (2021). "Structure and Hierarchy of SARS-CoV-2 Infection Dynamics Models Revealed by Reaction Network Analysis". In: *Viruses* 13.1, p. 14.

Peter, Stephan, Bashar Ibrahim, and Peter Dittrich (2021). "Linking Network Structure and Dynamics to Describe the Set of Persistent Species in Reaction Diffusion Systems". In: *SIAM Journal on Applied Dynamical Systems* 20.4, pp. 2037–2076.

Petri, Carl Adam (1962). "Kommunikationen mit Automaten". PhD thesis. PhD Thesis, University of Bonn.

Proctor, Carole J and Graham R Smith (2017). "Computer simulation models as a tool to investigate the role of microRNAs in osteoarthritis". In: *PLoS One* 12.11, e0187568.

Sensse, Anke and Markus Eiswirth (2005). "Feedback loops for chaos in activator-inhibitor systems". In: *The Journal of chemical physics* 122.4, p. 044516.

Smallbone, Kieran and Pedro Mendes (2013). "Large-scale metabolic models: From reconstruction to differential equations". In: *Industrial Biotechnology* 9.4, pp. 179–184.

Vazirani, Vijay V (2001). *Approximation algorithms*. Vol. 1. Springer.

Wagner, A and D A Fell (Sept. 2001). "The small world inside large metabolic networks". en. In: *Proc. Biol. Sci.* 268.1478, pp. 1803–1810.

7 Appendix

The supplementary data is available through Github by `https://github.com/WoitkeL/Master_thesis_supplementary_data`. If the branch is private, access can be given after a request to `linus.woitke@uni-jena.de`. There the following data will be available:

- the code of the functions described in the thesis, the code of the generation of figures and the validation of the data of kaleta, as well as a tutorial python file.
- The exel table which is the result of the analysis of the SBML files from the BioModels database
- the statistical analysis of the runtime of the pulp solver, including a pairwise correlation plot as well as a stepwise multiple regression
- the original SBML files of kaleta

8 Selbstständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Seitens des Verfassers bestehen (keine) Einwände die vorliegende Masterarbeit für die öffentliche Benutzung im Universitätsarchiv zur Verfügung zu stellen.

Unterschrift: _____

Jena, den 28.02.2023