



LUDWIG-MAXIMILIANS-UNIVERSITÄT
TECHNISCHE UNIVERSITÄT MÜNCHEN



**Wissenschaftszentrum Weihenstephan;
Lehrstuhl für Genomorientierte Bioinformatik**

Bachelorarbeit
in Bioinformatik

**Empirical reduction of
signaling networks**

Dennis Rickert

Aufgabensteller: Prof. Dr. Hans-Werner Mewes
Betreuer: Prof. Fabian Theis
Abgabedatum: 15.04.2009

Ich versichere, dass ich diese Bachelorarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

15.04.2009

Dennis Rickert

Abstract

Based on an cooperation with the group of Inna Lavrik at the DKFZ Heilderberg we assessed the requirements and constraints that are imposed on the automatic reduction of signaling models in an experimental context. We developed a set-based strategy of structuring the reduction process that allows a global analysis of the possible reductions of a model. Using Boolean logic we derived conditions that help us recognize invalid and redundant reductions on a logical basis without the need for simulation. The partial analysis of a large signaling model of the extrinsic apoptotic pathway allowed us to demonstrate the ability of our framework to handle models with a reduction space of $> 10^{10}$ possible reductions. Since computational limitations inhibited a complete analysis of the large model we used a smaller artificial model to demonstrate the possible analyses that can be performed based on our framework and their relevance for experimental groups.

Auf der Basis einer Kooperation mit der Gruppe von Inna Lavrik am DKFZ Heidelberg analysierten wir die Anforderungen und Einschränkungen welche für die automatische Reduktion von Signaling Modellen in einem experimentellen Kontext gegeben sind. Wir entwickelten einen Set-basierten Ansatz die automatische Reduzierung zu strukturieren um so globale Analysen eines models zu ermöglichen. Durch die Verwendung Boolescher Logik konnten wir Bedingungen entwickeln die uns dabei helfen logisch unmögliche und redundante Reductionen zu erkennen ohne sie zu simulieren. Die partielle Analyse eines großen Signaling Systems des extrinsischen Pathway der Apoptose erlaubte uns zu demonstrieren, das unsere Implementierung in der Lage ist Modelle mit einem Reduktionsspace $> 10^{10}$ möglichen Reduktionen zu analysieren. Da eine komplette Analyse des Experimentalmodels auf der Basis der zur Verfügung stehenden Rechenkraft nicht möglich war analysierten wir ebenfalls ein kleineres künstliches Model. Dieses Modell erlaubte uns die in unserem Framework möglichen Analysen zu demonstrieren und ihre Relevanz für Experimentalgruppen zu beurteilen.

Contents

1. Introduction	9
2. Evaluation of requirements imposed by experimental context	12
2.1. Biological background	12
2.2. Modeling techniques used for the CD95 model	15
2.3. Experimental restrictions to error estimation and parameter fitting	16
2.4. Unconfirmed reactions in the CD95 model	20
3. Overview of different reduction strategies	23
3.1. Timescale analysis	23
3.2. Quasi steady state approximation	24
3.3. Motif based approach to reduction	26
3.4. Set based approach to reduction	28
3.5. Comparison of different strategies	29
4. Basic definitions and implications of set based reduction	31
4.1. Set based reduction - an approach on two levels	31
4.1.1. Basic formalisms	32
4.1.2. Applying reductions to the biochemical model	33
4.1.3. Structuring the reduction space	34
4.2. Definition of basic reduction steps	35
4.2.1. Merging of states	35
4.2.2. Removal of reactions	36
4.3. The reduction graph - a supporting data structure	38
5. Standard algorithms for combinatorial optimization	41
5.1. Supported set based algorithm	41
5.2. Branch and bound algorithm	43
5.2.1. Last In First Out Branch & Bound	43
5.2.2. Priority Branch & Bound	44
5.3. Greedy search	46
6. Advanced reduction strategies	47
6.1. Boolean analysis of MA-network model	47
6.1.1. Controllability and observability	47
6.1.2. Identification of invalid reduction	49
6.1.3. Identification of redundant reductions	52
6.1.4. Implementation of Boolean model checking using BDD's	53

6.2. Heuristic strategies of priority estimation	54
6.2.1. Maximum information gain strategy	54
6.2.2. Analysis of prior fitted kinetic parameters	55
7. Application of the reduction algorithm	56
7.1. Analysis of the CD95 model	56
7.2. Analysis of a medium sized toy model	57
7.2.1. Model used	58
7.2.2. Methods	58
7.2.3. Results & Discussion	58
8. Discussion of results and further perspectives	62
Bibliography	63
List of Figures	66
A. External tools, packages and librarys used	67

1. Introduction

The reduction of biochemical models is a problem that, in a broad view, is significantly older than modern systems biology. The derivation of the Michaelis Menten approximation for enzyme kinetics from a multi step model of binding, unbinding and processing reactions over 90 years ago can be considered one of the first practical applications of biochemical model reduction.

Even one of the main reasons for performing model reductions stayed identical. The baseline model for enzymatic reactions prior to the work of Michaelis and Menten consisted of three separate reactions. Each of these reactions had its own kinetic parameter, so that the characterization of one enzymatic reaction required the determination of three parameters. To make things worse at least two of these parameters couldn't easily be determined in experimental setups as it usually wasn't possible to isolate the intermediate complex that was formed during the multiple step model.

Michaelis and Menten reduced the three steps model by analytical simplification of the basic equations that constituted the multiple step model, making certain assumptions regarding the parameters of the original model. This resulted in a way to characterize the behavior of an enzyme with only two parameters (the Michaelis Menten constant K_m and the maximum reaction speed v_{max}), both of which can be measured in simple experimental setups. The success of this method can be witnessed by the fact that the Michaelis Menten kinetic is up to this day considered the standard "textbook" model for enzyme kinetics.

With the advent of modern system biology the focus in biochemics shifted from the detailed analysis of single or few reactions to the analysis of large interacting networks. Modern experiments such as co-immunoprecipitation, DNA micro array analysis and the analysis of knockout-mutants will often be performed on a large scale, generating data for models that include dozens or even up to thousands of interacting species.

Creating quantitative models on the basis of such large amounts of data is one of the main challenges of system biology. Recent proceedings [1] and analyses "identified the lack of such quantitative information as a major barrier for the use of systems biology approaches to drug development" [2].

Unfortunately the experimental data available will often not suffice to create exact quantitative models, instead resulting in under determined models [3]. In under determined model some or even most parameters can have a wide range of values. This usually results in numerous alternative parameter sets that all explain the experimental data but make different predictions regarding the behavior for initial conditions that differ from the experimental setups.

This effect is even present in models where all reactions are confirmed with 100% accuracy and only their kinetic parameters are unknown [4]. This is similar to the situation

of Michealis and Menten 90 years ago; a complex model is characterized by a number of experimentally inaccessible parameters. In this situation model reduction is employed to reduce the number of parameters that characterize a complex model. Various standard approaches to the reduction of biochemical models exist; by making assumptions regarding such factors as the speed of certain reactions, steady states, equilibria or moiety preservation the number of parameters in the model is reduced.

This process does not in itself remove the problem of model under determination. The reduced models will usually still show alternative predictive behaviors. However clever model reduction often allows us to identify possible key parameters, e.g. a model that did depend on the parameters p_1 - p_5 prior to the reduction might now depend on only two parameters p_{red1} and p_{red2} . If the model reduction was performed with experimental conditions in mind p_{red1} and p_{red2} are both experimentally accessible, thus allowing us to focus future experimental setups on these key parameters.

The problem of model under determination becomes even worse if we consider models where not all reactions in the model are confirmed with certainty but some are only assumed to take place. While this seems an unlikely situation it actually occurs frequently. Sometimes it is possible to determine the substrate and the product of a reaction (for example by introducing radioactive carbon into the substrate as a marker) but not to identify the enzyme that catalyzes the reaction. Possible candidates for the catalyzing enzyme can be identified using co-localization by immunocoprecipitation.

In this cases the goal of model reduction will often be the identification of "minimal models", e.g. models that contain as few unconfirmed reactions as possible. Analysis of this minimal models might reveal key reactions among the hypothetical reactions. For example all minimal models derived from the initial model may contain either reaction 1 or reactions 2 and 3. Once such key reactions have been identified they can be used in the design of further experimental setups.

Biochemical models are used to model regulatory systems of numerous different properties. The size of models can range from a few dozen to thousands of states, chemical interactions can be modeled with various levels of detail, from boolean networks where a chemical species can either be present or not to the exact calculation of concentration values using differential equations. This has to be considered when talking about model reduction; for example a reduction strategy designed for boolean networks will often not be applicable to continuous systems.

While attempts have been made [5] to structure the engineering of biochemical models depending on these properties, we will often find a high degree of diversity in the way models are created. This is deleterious to the development of standardized reduction frameworks. Instead we will often find that, while standard approaches to model reductions exist their application to a biochemical model will require a significant amount of adaption. Both algorithmic (for example compatibility of data structures used) and logical (are the assumptions the standard approach involves applicable to the modeled system) problems have to be considered.

In this bachelor project we focus on this adaption part. In cooperation with the exper-

imental group of Inna Lavrik at the DKFZ in Heidelberg we try to create a framework that is suitable to perform automatic model reduction on one of the groups models. We consider the requirements imposed by their experimental context and the compatibility of model reduction standard approaches.

We will start by introducing a small formalism regarding the structurization of the mathematical models of biochemical system we intent to use. We consider a model to be composed of two components:

Definitions 1. *A models **topology** defines which reactions exist in a model. It contains the information which states are consumed in a reaction, which states catalyze a reaction and what the products of each reaction are. For a given system the topology suffices to create a mathematical model of this system. However this does not include the actual value of the kinetic parameters. This definition can be expanded to include the introduction of minimal and maximal values that a kinetic parameter can have, for example it could be a possible constraint that all kinetic parameters values k_n have to be between 1 and $1E-10$.*

*A models **parametrization** is the set of all its kinetic parameters. The kinetic parameters are numerical values; each kinetic parameter is assigned to one reaction. Without the topological information the kinetic parameters offer no informations regarding the model.*

Combining the information of a models topology with a possible parametrization results in a complete model that can be used to simulate the biochemical systems behavior for different initial conditions. This definition allows us to make a distinction between model reduction and model fitting.

- **Model fitting** is the process of changing the parametrization of a model. Usually model fitting will be used to change the models behavior to confirm with experimental observation. A wealth of standard algorithms (simulated annealing, differential evolution, parameter scanning and more) exist for this operation. It should be noted that model fitting is often computationally expensive.
- We will refer to the manipulation of a models topology as **Model reduction**. This will usually be done to simplify a biochemical model. It includes but is not limited to changing the kinetic (not the kinetic *parameters*) of a reaction, removing or replacing reaction and even removing or merging different chemical species.

Model fitting has no influence on a models topology, while changing a models topology will invalidate a models parametrization. Performing model reduction will usually require subsequent model fitting to the experimental data.

2. Evaluation of requirements imposed by experimental context

In this bachelor thesis we had to focus on the reduction of models with certain limited properties. This was required due to the numerous different ways biochemical systems can be modeled as mentioned in chapter 1. A limiting factor in the choice of model type was the availability of unreduced biochemical systems of this type. Most published models are already reduced, limiting their usefulness in evaluating a reduction framework.

We managed to enter into a cooperation with the group of Dr. Inna Lavrik at the DKFZ in Heidelberg. This cooperation gave us access to an unreduced version of a large biochemical signaling model and experimental data for this model. Personal conference with Nicolai Fricker and Dr. Inna Lavrik allowed significant insights into the possible use of a computational reduction framework for experimental groups, as well as the requirements such a framework would have to fulfill to be applicable in an experimental environment. In this chapter we will discuss these requirements as they profoundly influenced our work.

2.1. Biological background

The signaling system provided by Nicolai Fricker models the initiation of a cells apoptosis mechanism through the FADD or extrinsic pathway. The pathway is started by the binding of CD95 ligand, an secreted signaling molecule to the CD95 receptor, a receptor spanning the outer cell membrane. Binding of the ligand causes the formation of the intracellular membrane bound DISC (death inducing signaling complex). FADD is recruited to the cytosolic domains of the CD95 receptor and forms the basis of the DISC.

Further steps in the apoptosis pathway are less clear. The FADD part of the DISC is able to bind both procaspase 8 (called C8 in the model) and the two *FLIP* subtypes *FLIP_L* and *FLIP_S*. Experimental sources [6] suggest that the bound procaspase 8 is processed to the intermediate product p43/p10 and the final product p18 without dissociating from the DISC. This implies that the states CD95 FADD, FADD-FS, FADD-FL, FADD-C8, C8FS dimer, C8 homo/heterodimer and p43 homo/heterodimer are all bound to the DISC.

The DISC is assumed to have two binding sites, both of which can bind procaspase 8, *FLIP_L* and *FLIP_S*. *FLIP_L* has an unclear role in apoptosis, sometimes acting anti- and sometimes pro apoptotic. *FLIP_S* always acts anti apoptotic. This model formulates a possible hypothesis for these different roles. *FLIP_S* inhibits apoptosis by binding to the DISC (FADD-FS). Any DISC *FLIP_S* has bound to is unable to process bound procaspase 8 (C8FS-dimer). *FLIP_L* also binds to the DISC and can act inhibitive by depleting

the amount of DISC available for procaspase 8. If two molecules $FLIP_L$ bind to a DISC this DISC is effectively blocked from procaspase 8.

The pro-apoptotic effect $FLIP_L$ is assumed to occasionally display is explained by a favorable change in kinetics when a DISC bind each one molecule of $FLIP_L$ and procaspase 8, resulting in the C8heterodimer. The exact nature of the positive effect is however unknown. One possibility is that the procaspase 8 in the C8heterodimer is processed faster than procaspase 8 in the C8homodimer. Other possibilities include scenarios where the C8 heterodimer is not in itself faster processed, but recruits factors that help C8homodimers to be processed faster. All these possibilities are modeled by the wealth of possible regulatory interactions between the states C8 homo/heterodimer and p43 homo/heterodimer.

The rest of the model, e.g. the interactions between Parp, Bid and C3 are of a less regulatory nature. Parp, Bid and C3 are known downstream targets of p18. They are mainly included in this model since they are used as indicators of the onset of apoptosis in the experimental setup.

It should be noted that this model is only a simplification of the complete FADD signaling pathway. Other factors (like FAP-1, FLASH, RIP, Daxx and more) are known to bind to the DISC and have different influences. The number of downstream interactions of p18 was limited to those of experimental relevance. These simplifications are acceptable as long as we keep in mind that this model does only focus on the interactions of CD95L/R, $FLIP_L$, $FLIP_S$ and procaspase 8 and is no complete model of this apoptosis pathway. Significant changes to other signaling molecules that couldn't be modeled here might change the DISC based apoptosis signaling in a way that is not predicted by this model.

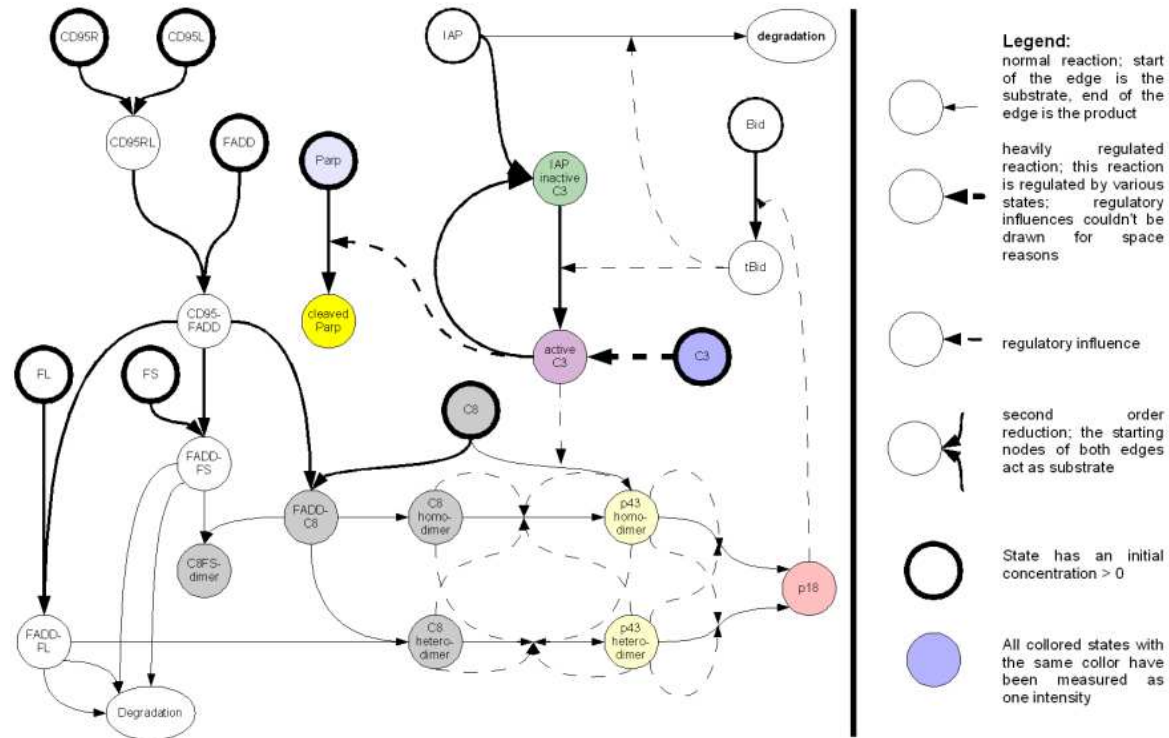


Figure 2.1.: Network representation of the CD95 model. It should be noted that the reaction that cleaves C3 and produces active C3 is potentially regulated by any C8 dimer, p43 dimer or p18; these regulatory influences could not be shown for reasons of illustrative clarity.

2.2. Modeling techniques used for the CD95 model

We already mentioned that different ways exist to model a biochemical system, such as different kinds of kinetics, simulation of concentrations using differential equations and boolean approximations. The apoptosis model discussed in section 2.1 was designed using mass action kinetics. Actual concentrations of chemical species were modeled using ordinary differential equations. As different ways exist to formally model mass action kinetics we will state the formal definitions we used:

- The model has a number of states S and reactions R . Each state represents the concentration of one biochemical species. Each reaction R is one biochemical reduction. All reactions are considered irreversible.
- The change of the states is described by a set of ODE's.
- The ODE's can be formulated in the following way:

$$d/dt(S_i) = \sum_{\forall k:r_k \in R_{prod}(i)} r_k(t) * n_k - \sum_{\forall j:r_j \in R_{cons}(i)} r_j(t) * n_j$$
- $R_{prod}(i)$ is the set of all reactions that produce S_i . $R_{cons}(i)$ is the set of all reactions that consume S_i . $r_i(t)$ is the reaction value of the reaction i at the time t . n_i is a stoichiometric parameter; e.g. if a reaction consumes or produces one molecule $n_i = 1$.
- The reaction value of a reaction i can be formulated as

$$r_i(t) = k_i * \prod_{\forall j:s_j \in S_{cons}(i)} S_j * \prod_{\forall l:s_l \in S_{cata}(i)} S_l$$
- $S_{cons}(i)$ is the set of all states that are consumed by reaction i . $S_{cata}(i)$ is the set of all states that catalyze reaction i but are not consumed. k_i is a kinetic parameter determining the speed of the reactions
- As an additional constraint only reactions of at most second order are allowed, this means $\|S_{cons}(i)\| + \|S_{cata}(i)\| = < 2 : \forall i$.

It should be noted that the model contained a large number of enzymatic reaction. These reactions have one state that is consumed and another that catalyzes the consumption, so for these reaction is $\|S_{cons}(i)\| = \|S_{cata}(i)\| = 1$.

Formulating enzyme kinetics in such a way using mass action kinetics is a valid approximation; it is usually employed when the Michaelis Menten parameters can't be estimated for technical reason or reasons of model complexity.

A model designed in this way can be simulated utilizing standard ODE solvers that are supplied by numerous mathematical programs like matlab; specialized packages like the SBtoolbox [13] we used offer a complete framework for the simulation of biochemical ODE-based models.

2.3. Experimental restrictions to error estimation and parameter fitting

The CD95 apoptosis model is subject to certain experimental restrictions regarding the measurements of the biochemical species involved. These restrictions have to be considered in the parameter fitting of the mathematical model. All of these restrictions are common in experiments in molecular biology.

All measured values, both for “input” and “output” states have a significant experimental error associated with them.

We will consider all states to be input states that have a concentration > 0 at $t = 0$; output states are all states that we measure experimentally. This also implies that a state can be both an input and output state. The problem of random experimental error is common to most experimental measurements, but especially common in experiments that involve measuring concentrations in living cells. Isolating and measuring specific molecules in cells requires a tedious process of breaking up cell walls and membranes, separating proteins, DNA, membrane fragments, small soluble molecules and other contents of the cell. If a protein is measured the next step will usually involve some kind of antibody selection like a western blot. Most of these steps, even if conducted with extremest caution will introduce some kind of error; this is considered unavoidable. Keeping this in mind is less important for modeling a biochemical system. Instead we have to consider this error when interpreting and comparing different parameter sets for a model and how well they explain the experimental data.

A common misconception often encountered is that a model that explains the experimental observations with 0% error has to be “correct” and preferred over a model that simulates a behavior that differs by 10% from the observed data. This is statistically speaking incorrect because of the error associated with the experimental data. Consider the following small example:

If we assume that the experimental error is Gaussian distributed (a common assumption when dealing with random error) with a standard deviation of 10%, we expect:

- 68% of all experimental measurements to have between 0%-10% error
- 27% of all experimental measurements to have between 10%-20% error
- 4% of all experimental measurements to have between 20%-30% error
- less then 1% of all experimental measurements to have $>40\%$ error

In this situation a common rule of thumb is to consider up to two standard deviations as an acceptable deviation. In this case we would decide that any model that explained the experimental data with an error of less than 20% is supported by the experiment. We would accept any model with less then 20% error and reject any with more but we wouldn't consider a model with 5% error better than a model with 15% error. This may seem trivial right now but will later be of importance when we discuss possible heuristics to determine reduction strategies; it invalidates most reduction strategies that try to evaluate the "error" of allready considered reductions to choose the next possible reduction.

For our considerations we will use an error cutoff of an relative error of 40%. This value was supplied by the group of Inna Lavrik as a cutoff they have used previously.

Some states can't be measured as isolated species but only together with other species.

It has already been mentioned that the experimental measurement of proteins is difficult. This isn't limited to the numerical error of measured concentrations. Common techniques like anti body identification of proteins or fluorescence analysis can result in false positive signals especially from structurally similar proteins. If markers are used to measure the concentration of a protein it will often be impossible to distinguish monomers and dimers of this protein. While more complex techniques like mass spectrometry will usually be able to distinguish even extremely similar enzyme it is often impossible to perform these on a large scale.

If a system contains different species that can't be measured separately with acceptable experimental effort these species will be treated as a single entity for the calculation of the error between the model and the measured data but still be modeled separately. This is the case for multiple proteins in the CD95 model:

- Caspase 8 is measured as a single entity, independently of its binding status. This includes unbound C8, Fadd-C8, C8-heterodimer, C8-homodimer and C8-FSdimer. The C8 homodimer contributes with double intensity, e.g. one mol C8homodimer is measured with the same value as two mol C8. Technically this measurement also include the C8 part of the procaspase 43 complex, however since procaspase 43 can be measured separately its contribution can be subtracted from the measured value.
- Procaspase 43 is measured as a single entity both with C8 homodimer and heterodimer bound.
- Once C3 has been cleaved it is impossible to distinguish between active C3 and C3 that has been inhibited by IAP.

The technical aspect of this problem is rather small; most toolboxes already support the ability to define "variables", e.g. weighted sums of different states that can simulate this kind of experimental constraint. However the ability to determine the behavior of a system using such combined states is diminished compared to the information gain that measuring each state individually would provide.

Measured values aren't absolute concentrations but rather relative values. Absolute concentrations are often unknown.

Another limitation of many experiments is the difficulty of determining actual concentrations. Often the experimental output will be a relative value that can be compared between different experiments. For some experiments standards exist that allow us to convert this relative values to absolute concentrations. In the cases of other proteins no such standard is available. Dealing with this problem usually involves the introduction of various new parameters.

If the initial concentration of a species is unknown this concentration is introduced as a new parameter that has to be fitted.

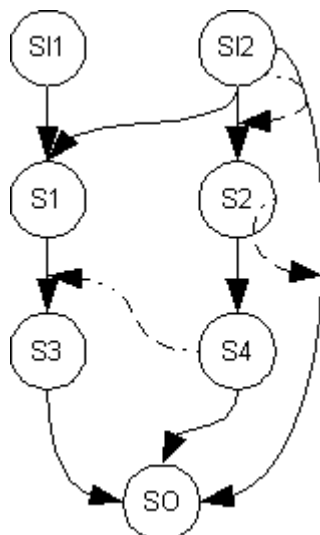


Figure 2.2.: Artificial model used to estimate the effect of initial condition fitting vs. kinetic parameter fitting

While this feature is easily implemented we found it to be a major problem in the actual fitting run. Treating initial parameters as fitable values¹ will often result in unwanted conflicts with the error function used. The initial conditions heavily influenced the behavior of the entire system and commonly produced fitting results that were stuck in local minimal with biologically nonsensical parameters.

A situation that commonly occurred when trying to fit the model parameters to the experimental data was that the initial randomized model tended to show "longer" periods of growth and more rapid growth than the experimental data suggested. This resulted in the difference between the simulated and the measured concentrations being greater than the value of the experimental concentration itself, leading to a difference of over 100%.

We will illustrate this using the small toy model illustrated in figure 2.2. We simulated an experiment using a small artificial model with 12 reactions, and 7 different states. The model contained two states (S1 and S2) with an initial concentration > 0 . This model was used to simulate experimental data. Then we randomized all kinetic parameters but kept the initial conditions constant and run another simulation. The initial simulation using randomized data was called "RandPara1". We started a fitting run using RandPara1 as initial parameter values. Some steps taken from this fitting run show how the parameters are slowly fitted to the experimental data. In contrast we also started a fitting run where all kinetic parameters were kept constant at the initial randomized values and only the initial concentration of S1 and S2 was changed. After three steps the fitting finished with the time series "reduced input".

¹It should be noted that if data of multiple experiments is available all initial condition have to be fitted in a way that keeps the relation between the experimental values. For example if two experiments have been performed where the initial concentration of state A in the second experiment was twice the concentration of state A in the first experiment, $A_1 = 100$ and $A_2 = 200$ is a valid fit of the initial conditions, however $A_1 = 100$ and $A_2 = 1000$ would be invalid!

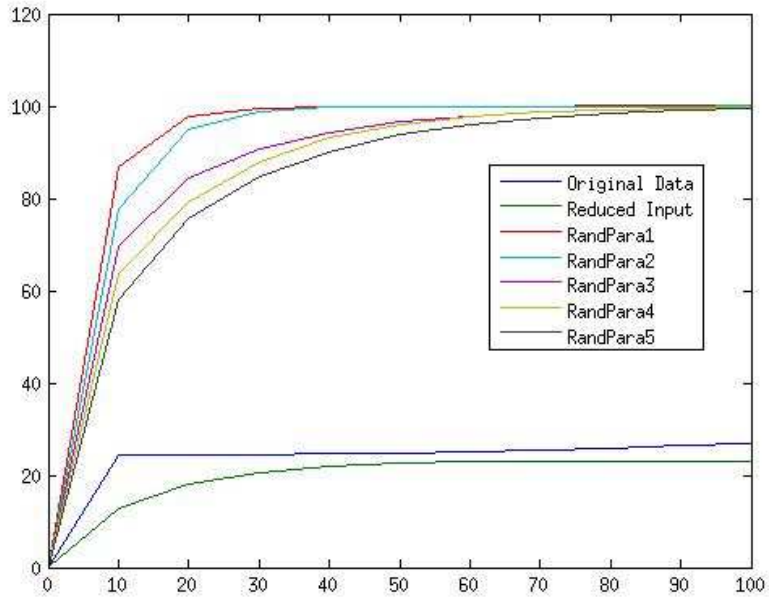


Figure 2.3.: Comparison of kinetic parameter and initial condition fitting

Plot of data generated using a toy model. Original data is the data generated using a toy model. RandPara1-5 are steps taking from a fitting run after the kinetic parameters of the toy model have been randomized. Reduced Input is a separate fitting run that used the same initial set of parameters as RandPara1, but instead of fitting kinetic parameters only the input concentration was fitted.

It should be stressed that "RandPara1" and "reduced input" both used identical kinetic parameters. RandPara1 used the correct concentration while the concentration "reduced input" was fitted to, was only 1/4 of the correct concentration.

Upon closer investigation the randomized parameters showed a behavior that, unlike the original models behavior was barely regulated. Instead the reactions that produced the measured output state proceeded unregulated at a high speed until the supply of S1 was completely consumed and converted into SO. The kinetic parameter based fitting, if observed over a complete fitting run did indeed reconstruct the regulatory connection. On the contrary a fitting run that allowed both the fitting of kinetic parameters and input concentrations (not shown) behaved identically to the "reduced input" run.

It should be noted that this kind of behavior could be suppressed using specialized error functions and adapted fitting parameters (simulated annealing using slow cooling schedules, high starting temperature and a large number of allowed failed fitting steps). While this reduced the problem of "input dominated fitting" it also increased the running time by a significant amount. We were not able to find parameters that allowed the fitting of the complete CD95 model with reasonable computational effort. More details regarding our adapted error function can be found in appendix [Ref einfügen].

If no standard exists that allows the conversion of measured "intensities" into absolute concentrations a normalization parameter has to be introduced to allow such a conversation. It will usually be possible to optimize this parameter using Linear Optimization.

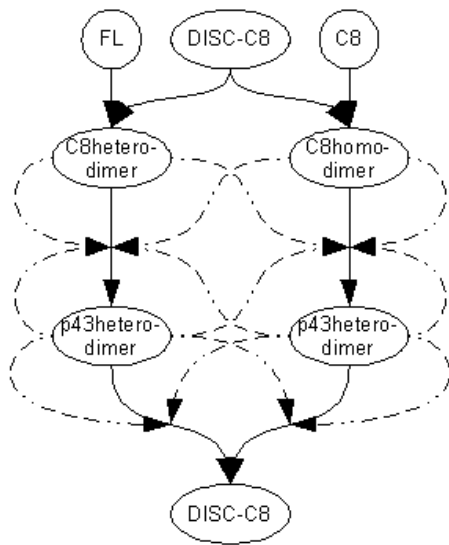
The optimization of this normalization parameter is performed during the calculation of the difference between the simulated model and the experimental data. This can be written formally:

- T is the set of all times a measurement was performed
- $s_i^{norm} \in \mathbb{R}$ is the normalization variable for State i
- $s_i^{exp}(t) \in \mathbb{R}$ is the intensity measured for State i at the time t
- $s_i^{sim}(t) \in \mathbb{R}$ is the concentration simulated for State i at the time t
- $error(a(t), b(t), T) \rightarrow \mathbb{R}$ is a formulation for a generalized error function, for example
- $error(a(t), b(t), T) = \sqrt{\frac{1}{|T|} * \sum_{t \in T} (a(t) - b(t))^2}$ is the RMSD
- $error(s_i^{exp}(t), s_i^{sim}(t), T)$ is the experimental error if no normalization is used
- $error(s_i^{norm} * s_i^{exp}(t), s_i^{sim}(t), T)$ is the experimental error if normalization is used
- to determine s_i^{norm} choose $s_i^{norm} = argmin(error(s_i^{norm} * s_i^{exp}(t), s_i^{sim}(t), T))$

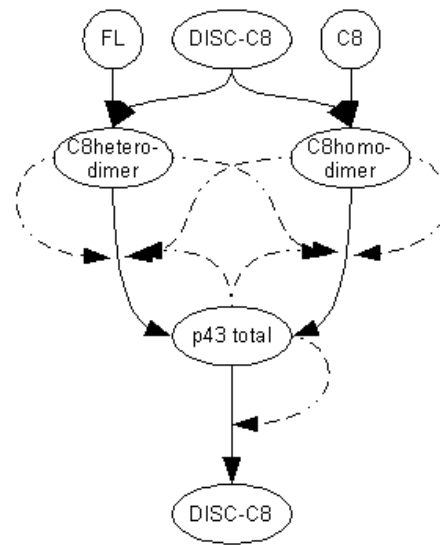
Since optimization over one parameter is implemented in most math environments the implementation of this normalization parameter is trivial.

2.4. Unconfirmed reactions in the CD95 model

The CD95 model contains a number of unconfirmed reactions. These are focused around the activation of procaspase 8 (labeled C8 in the mathematical model) and mainly result from the attempt to model the role of $FLIP_L$ (FL) in the CD95 apoptosis signaling pathway. Large amounts of $FLIP_L$ act anti apoptotic by inhibiting the CD95 activation pathway of apoptosis. The suggested mechanism for this is the blocking of the DISC(CD95FADD). The mathematical model of this mechanism is the binding of FADDFL to FL, depleting the amount of FADD available. Recent experiments suggest that smaller amounts of $FLIP_L$ actually act pro apoptotic and might be required in the CD95 signaling pathway. Knockout mice without any $FLIP_L$ expression will die during development with a phenotype reminiscent of procaspase 8 or FADD knockout mice. Further experimental evidence exist, but are outside of the scope of this thesis; more details regarding the experimental background of this mechanism can be found in [6]. Evidence suggests that a heterodimer of procaspase 8 and $FLIP_L$ bound to the DISC (C8heterodimer) will show enhanced activation of procaspase 8 compared to a procaspase



Subnet of the activation of procaspase 8 in the unreduced model with all possible unconfirmed reactions



Example for a possible reduction R1:
- the states p43 homo- and heterodimer have been merged

Figure 2.4.: Example for a possible reduction of a subnet of the CD95 model

8 dimer bound to the DISC (C8heterodimer). The mechanism for this enhancement is unknown; the mathematical model we analyzed contains reaction that model the possibilities that:

- C8homodimer and C8heterodimer auto catalyze their cleavage, resulting in the intermediate states p43homodimer / p43 heterodimer
- C8homodimer and C8heterodimer catalyze each others cleavage, resulting in the respective intermediate states p43homodimer / p43 heterodimer, e.g. C8homodimer catalyzes C8heterodimer vice versa.
- p43homodimer / p43 heterodimer catalyze the cleavage of C8homodimer.
- p43homodimer / p43 heterodimer catalyze the cleavage of C8heterodimer.
- p43homodimer / p43 heterodimer catalyze the cleavage of p43homodimer resulting in p18.
- p43homodimer / p43 heterodimer catalyze the cleavage of p43heterodimer resulting in p18
- active C3 catalyzes the activation of procaspase 8, resulting in p43homodimer.
- any form of partially activated C8 (C8homo/heterodimer, p43homo/heterodimer, p18) is a possible candidate for the cleavage of C3.

These unconfirmed reactions are the main focus of our analysis of the CD95 model. In its unreduced state it is very difficult to design experimental setups that help us to restrict the proposed reactions to a smaller number. A reduced version of the model could allow us a more focused experimental design. An example for a possible reduction could be to merge both p43 variants. The resulting model would have significantly less parameters that have to be fitted. An illustration of this reduction is given in figure 2.4. If this reduction would be supported by the experimental data further experiments could be designed to explicitly check this model and further validate it.

3. Overview of different reduction strategies

Since model reduction can be considered an established field of biochemical analysis various standard approaches exist. However while these established approaches are utilized with great success their application is usually limited to certain conditions. In the following chapter we will introduce two standard approaches to biochemical model reduction (timescale analysis and quasi steady state assumption) as well as two reduction strategies that we derived from other backgrounds (motif based reduction and set based reduction). We will evaluate their applicability to the requirements discussed in section 2.

3.1. Timescale analysis

Time scale analysis is a method based on the observation that while complex models will often have complex internal kinetic behavior this behavior is often "obfuscated" if only a few downstream nodes are directly observed. A common situation where this occurs results from a complex interaction of fast reactions followed by a very slow reaction. In other cases some reactions may follow a complex mechanism when observed for a long time but are almost constant during the time frame we observe.

An extreme example for these cases is the integration of different signaling pathways. Both the hormone pathway and neurological pathways show, if we consider their behavior separately, complex behaviors over time. However, if we model the response of a cell that integrates both hormone induced and neural signals to a neuronal stimulus we will find that the hormone induced part of the behavior will stay almost constant during the time frame we model.

But timescale analysis isn't limited, or in fact even focused on these extreme examples. Even reactions that differ only a few orders of magnitude in reaction speed can obfuscate the behavior of other reactions. Usually time scale analysis tries to classify every reaction in one of three classes depending on it's speed:

- slow reactions: Very slow reactions are considered to show an approximately constant behavior during the time frame we wish to model. Instead of creating ODE's for the reaction value of slow reactions their activity is modeled as a fix parameter.
- medium reactions: These reactions are modeled as normal.
- fast reactions: Are considered to happen instantly. This can be modeled in different ways. For example consider two states A and B that will always tend towards an

equilibrium where $B = 1.5 * A$. If the reactions that create the equilibrium are considered 'fast' they won't be modeled. Instead B will always be considered to be $1.5 * A$.

The classification required for this is non trivial. While very simple approaches may just consider kinetic parameters this usually isn't sufficient. Some reactions may appear to be slow but don't show a constant behavior during simulation. Some fast reactions can perform a regulatory function that is lost if they're considered instantaneous.

A considerable problem we expect to encounter while trying to apply an timescale separation approach to a model with the properties as discussed in chapter 2 is that the heuristics for timescale separation will often depend on kinetic parameters to be at least partially known. To utilize time scale analysis in this framework would require us to fit the parameters of the initial model to the experimental data.

Once this has been done timescale analysis could be performed to reduce any of the fitted models. The timescale separations performed in different fittings would then have to be analyzed. Both trivial (how often is reaction X classified as fast/slow) and non-trivial properties (if we classify reaction X as slow, how much will this simplify the model? If we consider reaction X as fast, does that influence how we can classify reaction Y?) could be analyzed. Based on these properties some kind of heuristic would have to be designed to find good reductions.

While this approach seems feasible a problem that we can't address in this way is the problem discussed in 2.4, e.g. that some reaction where only substrate and product are known for sure are included in different "versions" that use different catalyzing enzymes. Early fitting attempts revealed that these reactions are strongly under determined and the reaction parameters can have a wide range of values in different fitting runs ¹. In the worst case this could lead to a situation where most of the critical reactions are sometimes classified as slow and sometimes as fast.

3.2. Quasi steady state approximation

The quasi steady state approximation originates from the analysis of enzyme kinetics. It was first used by Briggs and Haldane [8] to formulate an enzyme kinetic that didn't depend on the assumptions made by Michaelis and Menten. The quasi steady state assumption in its classic application states that during an enzymatic reaction the absolute amount of enzyme substrate complex ES is generally constant, except for a very small time fraction after the start of the reaction (until equilibrium has been achieved) and the time when the substrate supply has been largely depleted.

We will use QSSA as an collective term for all methods that focus on making assumptions regarding the behavior of certain states or reaction to analytically simplify the

¹It should be noted that these fitting runs could not be performed with a satisfactory quality, therefore we can not state with certainty to which degree the reactions are under determined.

mathematical equations of the model, even if these assumptions do not necessarily involve steady states. We will use the derivation of the Briggs and Haldane kinetic to illustrate the application of the QSSA in model simplification.

In the following E is the concentration of an enzyme, S is the concentration of a substrate, ES is the enzyme substrate complex and P is the product of the enzymatic reaction. E_0 is the initial or total amount of enzyme, comprised of both unbound enzyme and enzyme substrate complex. Most classic approaches assume the substrate to be in massive excess of the enzyme, so that the consumption of substrate is less relevant to the speed of the reaction. Most enzymatic reactions are assumed to follow a simple scheme:

- The enzyme binds the substrate and forms the enzyme-substrate complex.
- The enzyme substrate complex can either dissociate without reaction (reforming enzyme and substrate) or react and form the product reform the unbound enzyme.
- The reaction is reversible until the product is formed. In almost all cases the creation of the product is considered an irreversible step.



Equations 3.1-3.3 formulate an enzymatic reaction as a sequence of three mass action reactions. This set of equations is the baseline for the approximation of most enzyme kinetics and is mostly considered the most accurate. It is also the most unwieldy, as it requires the measurement of three distinct kinetic parameters that aren't easily experimentally accessible (e.g. the inability to isolate a stabilized ES-complex without permanently changing the enzyme will make it impossible to estimate k_2 and k_3).

If certain relations between k_1 , k_2 and k_3 are assumed these equations can be used to derive the Michaelis Menten equation. The Briggs and Haldane assumption states that in most experimental setups $d/dt(ES) = 0$ will hold. This is the quasi steady state assumption, as it is assumed ES is in a steady state. We can use these equations to show:

$$\begin{aligned} d/dt(ES) &= +k_1 * E * S - k_2 * ES - k_3 * ES \\ 0 &= +k_1 * E * S - (k_2 + k_3) * ES \\ \text{since } E_0 &= E + ES \Rightarrow E = E_0 - ES \\ 0 &= +k_1 * (E_0 - ES) * S - (k_2 + k_3) * ES \\ k_1 * ES * S + (k_2 + k_3) * ES &= +k_1 * E_0 * S \\ (k_1 * S + (k_2 + k_3)) * ES &= +k_1 * E_0 * S \\ ES &= (k_1 * E_0 * S) / (k_1 * S + (k_2 + k_3)) \end{aligned}$$

$$ES = (E_0 * S) / (S + (k_2 + k_3) / k_1) \quad (3.4)$$

In 3.4 the term $(k_2 + k_3) / k_1$ can be treated as a single parameter, thus massively simplifying the enzymatic reaction. If we further assume $k_3 = 0$ we can simplify the Briggs

Haldane equation to the Michaelis Menten equation.

This general approach, despite being over 80 years old is still the foundation for modern QSSA. Various heuristic strategies exist to assume constant reaction rates or state values in biochemical models under different conditions, such as enzyme excess instead of substrate excess. Identifying states or reactions that can reasonably be assumed to be in a steady state will often allow us to analytically simplify the mathematical equations of a model by replacing several parameters with a single combined parameter. While this approach has been explored extensively in the past it is still subject to modern research (for example by [14]).

The main weakness in this approach is the identification of suitable steady state assumptions to simplify the equations compromising the model. Considering the signaling model system discussed in 2 is observed only in a time frame where we don't expect any kind of steady state or equilibrium most "classical" assumptions won't work.

A further possible deterrent to the application of an QSSA approach to our work is that especially more modern approaches will often add further assumptions regarding the relation of kinetic parameters that could be simplified. In an under determined network the kinetic parameters of multiple reactions might still be unclear to a degree that doesn't allow us to decide whether an assumption is valid.

An approach to create an automated reduction framework could try to heuristically check for states or reactions that could be assumed to be in some kind of steady state or equilibrium. Analysis of current different equilibrium and steady state definitions could be performed to derive algorithms that check a biochemical model for any subnets that could be assumed to fulfill some of these conditions. Once candidate states and reductions are identified some kind of heuristic evaluation would be employed to decide which assumptions we expect to provide the best model simplification to information loss ratio. These assumptions would then have to be taken into account to simplify the equations of the model. This kind of automated analytical simplification, while non trivial seems reasonably achievable, considering that symbolic math toolboxes (for example the Symbolic Math Toolbox for Matlab) provide a sophisticated framework for the automated manipulation of mathematic formulas.

3.3. Motif based approach to reduction

The general idea of this approach is to find recurring subnet motifs that occur with increased frequency in biochemical models. Once such motifs have been identified, we try to find smaller motifs that approximate the behavior of the original motif. A trivial example could be replacing the "motif" of a cascade with three steps with a single first order step.

The motif based approach to reduction shows a lot similarities to the QSSA. The difference is that the QSSA focuses on the assumptions that allows us to simplify the mathematical equations, while the motif based approach focuses on finding recurring mathematical regularities in the equations and then tries to make assumptions suitable

to simplifying these regularities.

Expanding this approach into an automated framework would at first involve integrating data of multiple biochemical networks into a common format and then search for recurring motifs. The motif definition could be derived by expanding the work of Milo et. al. [12] regarding network motifs to hyper graphs. Hyper graphs are a common network-based representation of biochemical models, where all substrates of a reaction constitute the start, and all products of a reaction constitute the end of a hyper edge. The behavior of the most prominent motifs could then be analyzed and hopefully assumptions could be found for some motifs that would allow analytical simplification of the equations that compromise the motif.

This approach has some potential problems. It isn't clear how well the motif concept could be expanded to hyper graph structures as the number of distinct motifs that are combinatorial possible grows significantly in respect to the number of nodes in a motif for hyper graphs than for normal graphs.

If we only consider reactions of at most second order, not allowing any self loops the number of reactions in a hyper graph is

- first order reactions: $N * (N - 1)$
- second order reactions: $(N * (N - 1))/2 * (N - 2)$ (where $(N * (N - 1))/2$ is the number of all pairs of start nodes of a second order hyper edge)

As we can see this grows by one potency faster than the number of reactions in normal graph (which is comprised of only the first order reactions), thus potentially generating more possible motifs even for few nodes. Since all possible motifs have to be counted in a large number of networks this could result in a significant increase in computational time required.

Another problem is that Milo et.al. employed a sophisticated randomization algorithm to compare the occurrence of motifs in real networks to the number of occurrences in random networks. Unfortunately the randomization of hyper graphs is a field that has been explored a lot less than the randomization of normal graphs for which extensive work exists (for example the graph evolution model by Erdős and Renyi [9]) regarding how to keep properties such as degree distribution invariant in randomization.

However the largest problem we encountered in our initial exploration of this field was the unavailability of integrated data for non reduced biochemical models in a data format that allowed the application of graph based motif search.

Sites such as biocyc [15] and reactome.org [16], while using standardized exchange formats focus on models that have already been reduced pretty thoroughly. These models are obviously unsuited for searching motifs that are common in unreduced models.

Models that have not or only partially been reduced are mainly available directly from publishers. While some models can be found in this way, they show a significant inhomogeneity regarding to the format used. We estimated the effort to integrate enough of

these models into a common data format to run motif searching algorithms on them to be to large to be accomplishable in the scale of this work.

3.4. Set based approach to reduction

This approach has been less formally explored than those we discussed up to now. By set based reduction we suggest that the reduction should be structured in a way that allows the application of standard set based optimization algorithms such as discrete branch and bound algorithms and greedy search. Ideally we would want the following conditions to be fulfilled:

- Reductions are compromised by elementary reduction steps (for example removing a reaction from a model might be an elementary reduction). These steps are handled in a discrete fashion, i.e. can either be performed or not performed but do not depend on continuous parameters.
- The number of possible reductions for a given model is finite.
- No elementary reduction may depend on another reduction to be applicable to a model.

In such a framework the reduction problem has the following properties:

- While any set of elementary reduction steps is a possible reduction not every reduction is supported by experimental evidence.
- The challenge lies in identifying which reactions are NOT supported by experimental data.
- Brute force checking of every reduction will usually not be possible.
- Heuristic strategies need to be employed to speed up the identification of not supported reductions.

In such a framework the total number of possible reductions of a model may not exceed $2^{N_{poss}}$ where N_{poss} is the number of possible elementary reduction steps for the model. This reduction space contains all reductions for a model under a given definition of elementary reduction steps. While the reduction space is of exponential size we believe it might be reasonable to assume we can apply heuristic strategies and clever structurization of the reduction space to explore it in more reasonable time.

These heuristics do obviously depend on the definition of elementary reductions utilized. For example if we consider removing a reaction from the model an elementary reduction we can limit the reduction space we need to explore by automatically invalidating all combinations of elementary reductions that completely disconnect an output node. This and other examples will later be discussed in chapter 6.

This approach offers several advantages. The definition of a clear reduction space allows us to perform reductions in a less subjective way. As an example this framework makes it easy to define a "smallest" model supporting some experimental data as the model that results from the application of the most elementary reduction steps. This allows us to justify our reduction based on the principle of Ockhams Razor. Of course this kind of reasoning does only hold within the defined elementary reduction framework. It is therefore necessary to define elementary reductions in a way that allows us to reproduce the reductions that experimental groups perform now on an expert knowledge basis.

In a way this characterizes the set based reduction approach as a "meta-strategy". It might be reasonable to define a framework where the elementary reduction steps consist of timescale analysis or QSSA analysis steps. However such endeavors, while possibilities for future projects seem outside of the scope of this bachelor project.

A set based approach implemented in the limited time of this project would have to focus on small elementary reductions without complicated side conditions. Analyzing the properties of the reduction space resulting from a simple definition of elementary reductions to allow exploration of the reduction space would be a main focus of this kind of project. The set based organization of the problem would allow us to employ standard set based algorithms and evaluate their behavior in the reduction space.

3.5. Comparison of different strategies

We will start our analysis of the different approaches considered with a short summary of advantages and disadvantages of the different strategies:

time scale analysis:

- + established method
- + implementations of standard algorithms are available
- classification of different timescales in under determined networks seems problematic
- doesn't address the problem of uncertain network topology

QSSA:

- + established method
- + strong mathematical foundation
- + the mathematical assumptions applied to a successful reduction might be used to derive hypotheses about biological implications (e.g. interpreting an equilibrium in the model in a biological context)
- standard assumptions can be expected not to hold due to model acting far away from steady state

- more advanced assumptions might depend on properties that can't be checked due to under determined state of network
- doesn't address the problem of uncertain network topology

Motif based approach to reduction:

- + statistical analysis allows focusing on relevant motifs e.g. motifs expected to be found in many models
- + can be based on prior work regarding network motifs
- expanding the idea of graph motifs to hyper graph motifs can result in various problems (few models exist for hyper graph randomization, number of possible motifs for a given number of nodes grows very fast)
- the number of integrated non reduced models required for a statistically relevant analysis of over representation of certain motifs seems unobtainable

Set based approach to reduction:

- + set based approach "structures" the space of possible solutions and makes it easier to determine properties such as how "small" a model is and if a smaller model could exist
- + standard algorithms for set based optimization problems can be utilized to have a foundation for possible heuristics
- +/- can be considered a meta method focused on more structured analysis
- depends strongly on finding suitable heuristics
- in the worst case the space of possible reductions can't be reduced from exponential size

As we can see both classical approaches don't address the specific problem of uncertain network topology. While it might have been possible to reduce some parts of the network using these approaches the information gain we expected about the reactions with uncertain catalyst was pretty small.

The motif based approach was excluded as a possibility pretty fast; the technical obstacles in integrating different non reduced networks from the various data structures provided by prior publications was deemed prohibitive.

Based on this considerations we decided to try and implement a set based reduction framework. We planed to focus on a simple elementary reduction to demonstrate the general concept of using a set based problem definition to limit the space of possible solutions in a structured way.

4. Basic definitions and implications of set based reduction

In our overview of different reduction strategies we characterized set based reduction as a meta strategy. We will now take a closer look on this characterization and consider its implications. A core idea of this approach is to define elementary reductions, e.g. reductions that can be applied without the need of further continuous parameters. Any reduction may either be applied to a biochemical system or not; however if it is applied its effect on the systems topology should be a strictly defined change of the system that does not depend on further random factors or parameters. Please note the distinction made in definition 1; while the model topology should be clearly defined the models parametrization is still subject to random factors. We will also demand that any set of reductions applied to a model will always have the same effect on the models topology, independently of the order in which the reductions are applied.

It should be noted that it is possible to define elementary reductions with discrete parameters, if the space these parameters can be taken from is limited using generalized logical formalisms as defined in [10]. A possible idea for the definition of an elementary reduction might be to constrict a specific kinetic parameter to a range of certain parameter values; for example we might want to constrict parameter p_1 so that $0.5 \leq p_1 < 0.6$. We could define a set of basic elementary reduction prior to the start of the reduction algorithm $k \in [1 : 10]$, $R_k(i) =:$ constrict the parameter p_i to a value $(k-1)*0.1 \leq p_i < k*0.1$. This set of elementary reduction is obviously finite. Applying $R_6(1)$ would result in the desired reduction.

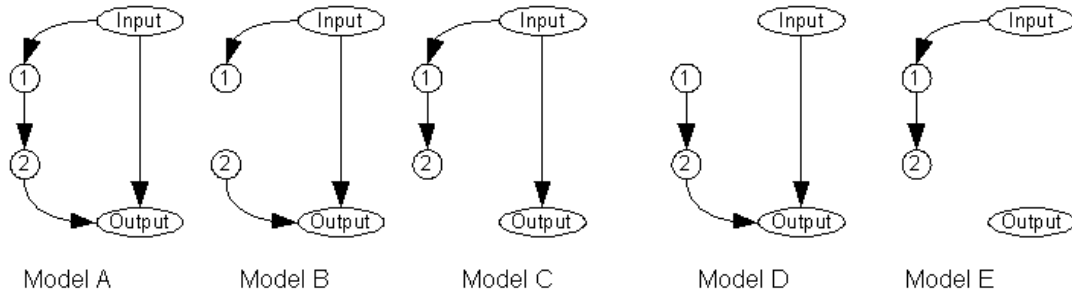
4.1. Set based reduction - an approach on two levels

The set based definition of elementary reactions will often contain redundant and/or obviously¹ invalid reductions. The exact nature of these redundancies and invalidities depends on the exact definition of the elementary reductions. A small example of redundant and obviously invalid reductions is given in figure 4.1.

A possible approach to this problem is considering it on two interacting but generally separate levels. One level is the the structurization of the "reduction space", e.g. the set of

¹It should be clarified that we'll call a reduction obviously invalid if it can be shown without running a fitting algorithm that it is impossible to fit the reduction to the given experimental data; it should not be misunderstood to imply that this proof is "simple" or "trivial"

all possible reductions that can be derived by combining any number of elementary reductions. The second level is the actual application of a reduction to the biochemical model. The structurization part is focused on identifying redundancies, while the application of reductions focuses on efficient checking for invalid reductions.



Model A is the original model. Only the state "output" is measured. Model B and C are redundant; both prevent that output is formed by consuming species 2; model B prevents the formation of species 2 while model C lacks the reaction that converts species 2 to output. Note that the behavior of B + C is different from Model D; in Model D all input will eventually be converted to output, while in models B+C some input can be "drained away". This illustrates that checking for redundancy has to be implemented carefully. Model E is obviously invalid if any measurement shows the formation of output, since no reactions remain that could produce output.

Figure 4.1.: Illustration of the concepts of redundancy and obvious invalidity

4.1.1. Basic formalisms

We will now give some basic definitions to better illustrate the different levels.

R_{ele} is the set of all elementary reductions.

$R_{complete} = \{\forall r_k : r_k \subseteq R_{ele}\}$ is the set of all subsets of the set of all elementary reactions.

$evaluate_{Model, ExperimentalData} : R_{complete} \rightarrow [0, 1]$

is an evaluation function. Given the model of an experimental system and experimental data it evaluates whether it is possible to reduce the initial model and still fit it to the experimental evidence.

$R_{valid} = \{\forall r_k \in R_{complete} : evaluate_{Model, ExperimentalData}(r_k) = 1\}$

is the set of all valid reductions.

$R_{invalid} = \{\forall r_k \in R_{complete} : evaluate_{Model, ExperimentalData}(r_k) = 0\}$

is the set of all invalid reductions.

$fit_{ExperimentalData}(Model.topology) \rightarrow Model.parametrization$

is a standard fitting function that tries to find a parametrization for a given model topology that explains some experimental data

$model_error_{ExperimentalData}(Model) - > \mathbb{R}_{\leq 0}$

is a standard error function (for example RMSD) that calculates the difference between the simulated data and the experimental data.

4.1.2. Applying reductions to the biochemical model

This "level" of the set based reduction approach is mainly focused on an efficient implementation of the evaluation function. While a trivial implementation can be achieved simply enough by combining standard error- and fitting functions (see Pseudocode 1, `simple_evaluate`), the performance of such an implementation often won't suffice. The evaluation function should ideally implement a efficient way of checking whether a reduction is obviously invalid.

```
function simple_evaluate(Reduction, Model, ExperimentalData, ...
    qualityCutoff, maxAttempts) = {
    new_topology = apply Reduction to Model.topology
    for (i = 1:maxAttempts)
        new_parameters          = fit(ExperimentalData, new_topology)
        currentModel            = new Model();
        currentModel.topology    = new_topology
        currentModel.parametrization = new_parameters
        currentError             = model_error(ExperimentalData, currentModel)
        if (currentError <= qualityCutoff ) return 1; endif
    endfor
    return 0;
}

function improved_evaluate(Reduction, Model, ExperimentalData, ...
    qualityCutoff, maxAttempts) = {
    new_topology = apply Reduction to Model.topology
    for (i = 1:maxAttempts)
        new_parameters          = fit(ExperimentalData, new_topology)
        currentModel            = new Model();
        currentModel.topology    = new_topology
        currentModel.parametrization = new_parameters
        if checkModelIsObviouslyInvalid(currentModel)
            currentError = ∞
        else
            currentError          = model_error(ExperimentalData, currentModel)
            if (currentError <= qualityCutoff ) return 1; endif
        endif
    endfor
    return 0;
}
```

Pseudocode 1: Example for a simple and an improved version of the evaluation function

Checking whether a model is obviously invalid is no trivial task. Various concepts derived from the basic analysis of nonlinear systems like observable and influenceable states can be used to formalize a framework that helps with this decision, an approach utilized by [17]. In chapter 6.1 we will discuss the implementation of an checking algorithm.

4.1.3. Structuring the reduction space

Based on the definition of the evaluation function we gave, we managed to structure the reduction problem in a way that allows us to formally characterize it as a satisfiability problem on a limited number of boolean variables. Each variable represents a possible elementary reduction. A reduction composed of different elementary reductions is an assignment of truth values to all boolean variables, where the variables that compose the reduction are *true* and the variables of all reductions that are not performed are "false". If a reduction is possible for the given experimental data we consider the underlying boolean function satisfied for this assignment.

The underlying boolean function is inaccessible to us and depends on the initial model, the experimental data and the possible elementary reductions. The size of the truth table of the boolean function is $2^{\|R_{ele}\|}$, and the set of all satisfying assignments can be mapped on R_{valid} . We will call this unknown boolean function $eval_{boolean} : [0, 1]^{\|R_{ele}\|} \rightarrow [0, 1]$. It should be obvious that $evaluate_{Model, ExperimentalData}$ and $eval_{boolean}$ are functionally identical except for different data structures. Since each naive checking of a possible reduction involves a call of the evaluation function and thus a computationally expensive fitting run, simply checking the entire truth table does not seem to be a valid strategy.

We propose the solution to define further boolean clauses that depend on the structure of the elementary reductions. These should be structured in a way that we expect any assignment satisfying assignment to also satisfy these additional clauses.

As a small example assume:

- A model that contains the reactions A, B, C, D, E
- The elementary reductions R_A, R_B, R_C, R_D, R_E . The reduction R_A is constituted by removing reaction A from the model and so on.
- A fitting framework that is able to fit a kinetic parameter to the value zero.

In such a situation we might reasonably assume that it is impossible to create a valid reduction from an invalid reduction that couldn't be fitted to the experimental data by performing another elementary reduction / removing another reaction. This should be obvious: removing a reaction is identical to setting its kinetic parameter to zero, so the fitting algorithm could have achieved this without the need for further reductions.

Based on this assumption we find that:

Any invalid reduction r_k invalidates all reductions r_j where $r_k \cap r_j = r_k$, e.g. any reduction that contains all elementary reductions in r_k is invalid no matter which further reduction it contains.

Any valid reduction r_k validates all reductions r_j where $r_k \cup r_j = r_k$, e.g. any reduction that contains only a subset of the reductions in r_k is also valid.

Utilizing this conclusions we're able to determine to multiple elements of the truth table of $eval_{boolean}$ performing a single evaluation process. For example we might check the reduction $\{R_A, R_B, R_C, \}$ and find it valid; this also validates the reductions $\{R_A, R_B\}$, $\{R_A, R_C\}$, $\{R_B, R_C\}$, $\{R_A\}$, $\{R_B\}$ and $\{R_C\}$.

The utilization of rules like these allows us to potentially explore exponential reduction space in acceptable time. Utilizing standard search algorithms for set based / boolean problems and maximizing the information gain from each validation step seems a plausible strategy to completely analyze the potential reductions of a biochemical model.

4.2. Definition of basic reduction steps

We already established that both conceptual levels of the set based reduction depend strongly on the definition of the possible elementary reductions. In theory various established strategies like time scale analysis could be adapted; for example it seems feasible to adapt a time scale analysis algorithm to a set based reduction strategy by defining the classification of one reaction as either fast or slow as two possible reductions (performing the classification as both fast and slow would automatically lead to an invalid reduction). We considered several approaches, but in the end decided to opt for a "simple" reduction strategy to demonstrate the general strengths of the set based reduction approach. Based on the non automated reduction strategies that had previously been employed by the group of Dr. Inna Lavrik we decided to consider the removal of reactions and the merging of states as possible reduction candidates.

4.2.1. Merging of states

A possibility we considered was a reduction that merged two states. This can best be explained based on a network / hypergraph interpretation of the biochemical model, where each state represents a node in the network graph and each reaction is a directed hyper-edge. The classic definition for the merging of two or more nodes is achieved by removing all merged states from the model and replacing each reference to any of them with a reference to a new state. Three topologically different situations can occur if two states are merged (these situations, along with some problems that can result from merging are illustrated in figure 4.2):

- The states are *parallel*, the network does not contain any directed path that contains both states.

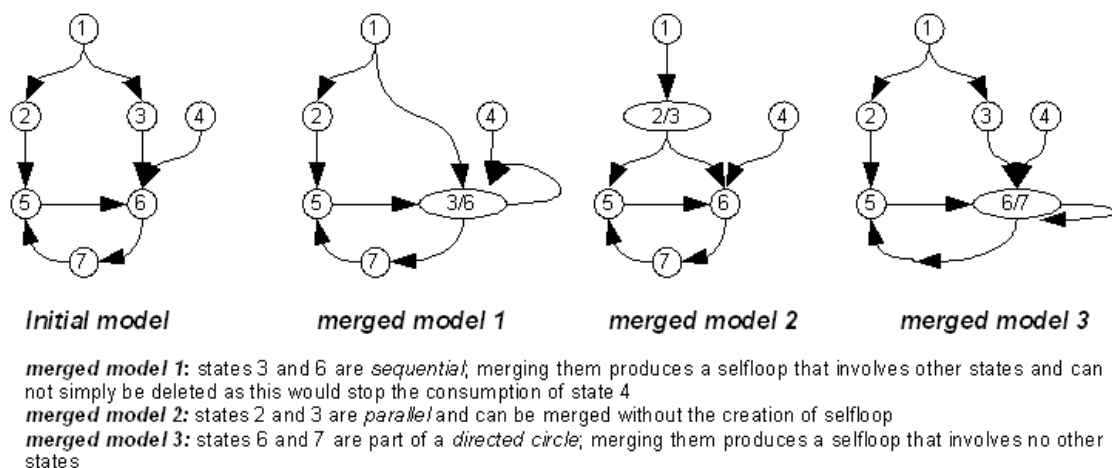
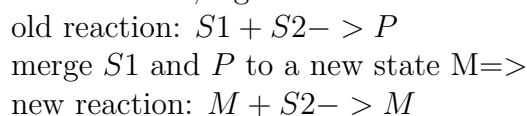


Figure 4.2.: Illustration of the different topological relations states can have and of possible problems that can result from merging states

- The states are *sequential*, every directed path in the network that passes both states will always pass them in the same sequence.
- A *directed circle* exists that passes through both states.

Merging states can result in problems when a self loop or circle is introduced to the graph if none was present prior to the reduction. While the reduction created in such a way may still be able to explain the experimental data its biological implication will often be obscured; the introduction of new self loops into a model that contained none previously will seldom make sense from a biological point of view. Unfortunately the classic definition of merging nodes (derived from the merge operation in normal graphs) will introduce a self loop if the product of a second order reaction is merged with one of its substrates, e.g.:



This forced us to abandon the classical merge operation for one that more suitable to handle hypergraph topology. We tried various variants (for example only allow the merging of the product of a reaction with its substrates if all its substrates are merged) but found no strategy that avoided reductions that invalidated biological background while still being widely applicable to the network we tried to reduce.

4.2.2. Removal of reactions

One of the simplest and most illustrative reductions possible is the simple removal of a reaction from a biochemical model. It has the additional benefit of producing reductions that can easily be interpreted in a biological context; any reaction removed is assumed

either not to take place or at least not to be a determining factor of the systems behavior. The implementation of this reduction is equally simple; setting the kinetic parameter of a reaction to zero has the same effect as removing the equation of the reaction from the ODE system.

Assumption of perfect fitting

For the design of our general concept we assume that every reduced model for that a parametrization exists that explains the experimental data will indeed be confirmed by the fitting algorithm employed. Without this assumption we would not be able to gain any information from invalidating a reduction, requiring us to perform a evaluation process for every reduction that is in fact invalid. During a "proof of concept" stage we considered such an assumption to be ok. Later application of our strategy would require further validation of this assumption. We will provide a short outline for such a validation:

- Consider all reductions that have been validated by the parameter fitting algorithm.
- Calculate the probability for a false negative, e.g. that one fitting attempt of a fittable reduction fails to fit the parameters
- Calculate the probability that n fitting runs of a valid reduction all produce false negatives where n is the number of fitting attempts that have been performed for each reduction
- Calculate the probability that one or more of the invalid reductions are in fact false negatives where every fitting attempt failed.
- Use this probability to either validate the assumption or re-run the simulation with a greater number of fitting attempts.

Logical validation and invalidation rules

The rules for logical validation and invalidation in a framework based on reaction removal have already been stated and explained as an example in section 4.1.3. We will therefore only restate them here as theorems and refer to the corresponding section for more explanations.

1. Any invalid reduction r_k invalidates all reductions r_j where $r_k \cap r_j = r_k$, e.g. any reduction that contains all elementary reductions in r_k is invalid no matter which further reduction it contains.
2. Any valid reduction r_k validates all reductions r_j where $r_k \cup r_j = r_k$, e.g. any reduction that contains only a subset of the reductions in r_k is also valid.

Maximal valid reductions and minimal invalid reductions

Based on the prior section we find that most states can (in an ideal case) be validated without fitting them. However some states have to be fitted as no other state either validates or invalidates them. These states can be divided in two classes:

Minimal invalid reductions are invalid reductions that become valid if any one of the elementary reduction that compose this reduction is not performed.

Maximal valid reduction are valid reductions that become invalid if any one further elementary reduction that hasn't already been performed is performed.

Knowing all minimal invalid and maximal valid reductions completely defines the space of possible reductions. This is useful in various applications. For example if we want to validate the assumption regarding perfect fitting we don't have to consider *all* reductions we performed during the model reduction but only the minimal invalid and maximal valid reductions. Other applications are more of a practical nature and focused on the implementation. Since the reduction space grows exponentially with the number of possible reductions the datastructure that is utilized to save the validated and invalidated states can become very large. Minimal invalid and maximal valid reductions offer a smaller representation of the reduction space if actions like comparing different reduction spaces are performed.

4.3. The reduction graph - a supporting data structure

We will now introduce the reduction graph as a supporting data structure. The reduction graph is a hierarchical graph. For an initial model with $\|R_{ele}\|$ elementary reduction it has the following properties:

- It has $\|R_{ele}\| + 1$ hierarchical levels.
- Each level has $\binom{\|R_{ele}\|}{n-1}$ nodes, where n is the number of the hierarchy of the current level.
- Each node is labeled by n-1 different elementary reductions, no two nodes are labeled in the same way.
- This essentially means that the n-th level contains all reductions that are composed of n-1 elementary reductions.
- The root contains the unreduced model.
- Let $Label(v_j)$ be defined as the set of all reduction that node j is labeled with.
- Two nodes v_1, v_2 are connected if $\|Label(v_1) \cap Label(v_2)\| = 1$.
- If two nodes are connected the node with the smaller number of labels is called the parent, the node with more labels is the child.

- A node v_1 is called the ancestor of v_2 if:
 $Label(v_1) \setminus Label(v_2) = \emptyset \wedge Label(v_1) \cup Label(v_2) = Label(v_2)$
- A node v_1 is called v_2 's descendant if v_2 is the ancestor of v_1
- In addition to its label each node has one of three states: valid, invalid or undetermined

The reduction graph has a total of $2^{\|R_{ete}\|}$ nodes, each of which represents one possible reduction. The three states represent our knowledge of the reduction space. When we begin a reduction most of the reduction graph will be in a "undetermined" state. As we evaluate different reductions states are changed to either valid or invalid

The reduction graph is in all algorithmic concerns equivalent to the prior boolean formulation; however it allows significantly more illustrative description of the implemented algorithms. For example let us consider the state validation and invalidation rules defined in section 4.2.2:

Any invalid reduction r_k invalidates all reductions r_j where $r_k \cap r_j = r_k$, e.g. any reduction that contains all elementary reductions in r_k is invalid no matter which further reduction it contains.

Can be implemented as :

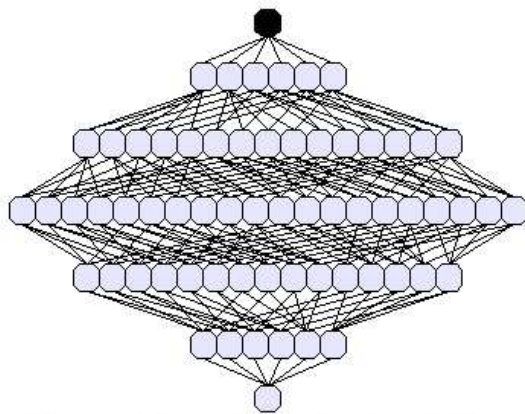
Whenever you mark a node as invalid, also mark all its descendants as invalid

Any valid reduction r_k validates all reductions r_j where $r_k \cup r_j = r_k$, e.g. any reduction that contains only a subset of the reductions in r_k is also valid.

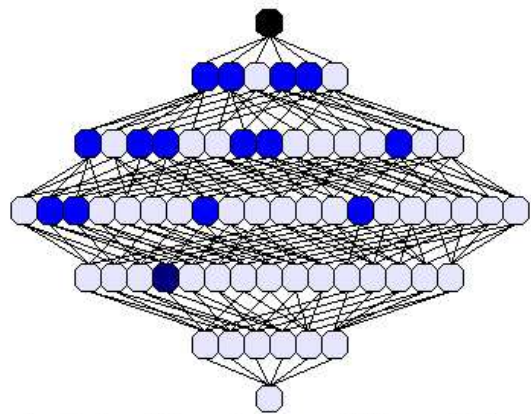
Can be implemented as :

Whenever you mark a node as valid, also mark all its ancestors as valid

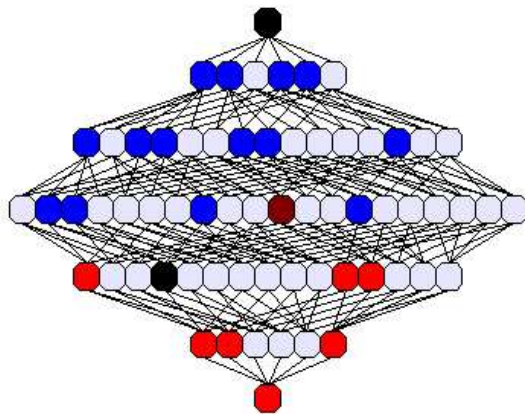
Both rules are visualized in figure 4.3



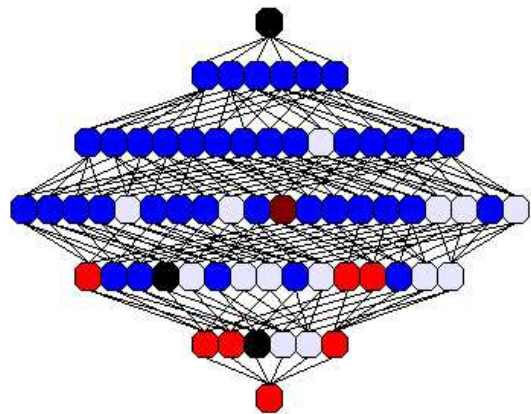
Initially only the root (unreduced model) has a known status, the rest of the reduction graph is unknown



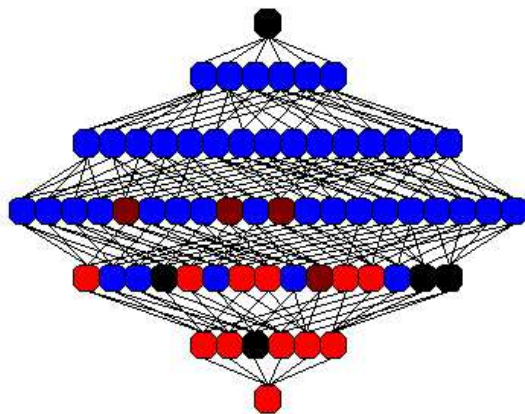
A first valid reduction validates all its ancestors



A first invalid reduction invalidates all its descendants



The evaluation of further reductions allows us to assign a known (valid or invalid) status to most nodes



The complete reduction graph is in a known status, the analysis is complete

Legend:






-  unknown reduction state
-  valid reduction, was not fitted but could be validated indirectly
-  valid reduction, was fitted
-  invalid reduction, was not fitted but could be invalidated indirectly
-  invalid reduction, was fitted

Figure 4.3.: Illustration of validation and invalidation in the reduction graph

5. Standard algorithms for combinatorial optimization

5.1. Supported set based algorithm

This algorithm is the adaptation of a standard frequent item set mining algorithm. Frequent item set algorithms are generally run on a large collection of item sets. The goal is to find subsets of items which occur frequently together in different item sets. To illustrate this idea better consider the following sets:

Set 1 = { A, C, D, F}

Set 2 = { A, B, D}

Set 3 = { A, C, D, F}

Set 4 = { C, D, E, F}

Set 5 = { A, C, D}

The item set {C, D} would have a support of four as the Sets 1, 3, 4, 5 each contain the subset {C, D}. Likewise the Set {C, D, F} would have a support of three.

Frequent item set mining algorithms aim to find all subsets that occur at least a certain number of times together. Usually the size of the collection of sets will cause the operation of checking whether any given item set is frequent to be computationally expensive, resulting in the need of efficient heuristics. One such heuristic is employed by the Apriori algorithm as introduced by Agrawal et. al. [11]. This algorithm is optimized to find all frequent item sets in a collection of sets while checking as few non-frequent item sets as possible.

The general idea of this algorithm is that if a item set of N elements is frequent then each subset of (N-1) elements of this item set also has to be frequent. This is very similar to the assumption we made in chapter 4, namely that if it can be shown that a reduction that removes a certain set of reactions can be fitted to support the experimental data then it's also possible to fit any reduction that removes a subset of these reactions.

A further common point between both problems is that checking one set for frequency of occurrence / one reduction for support of experimental data is an expensive operation. These similarities allow easy adaptation of the Apriori algorithm; the only change we have to perform is to check whether removing a set of reactions is supported instead of checking whether an item set is frequent.

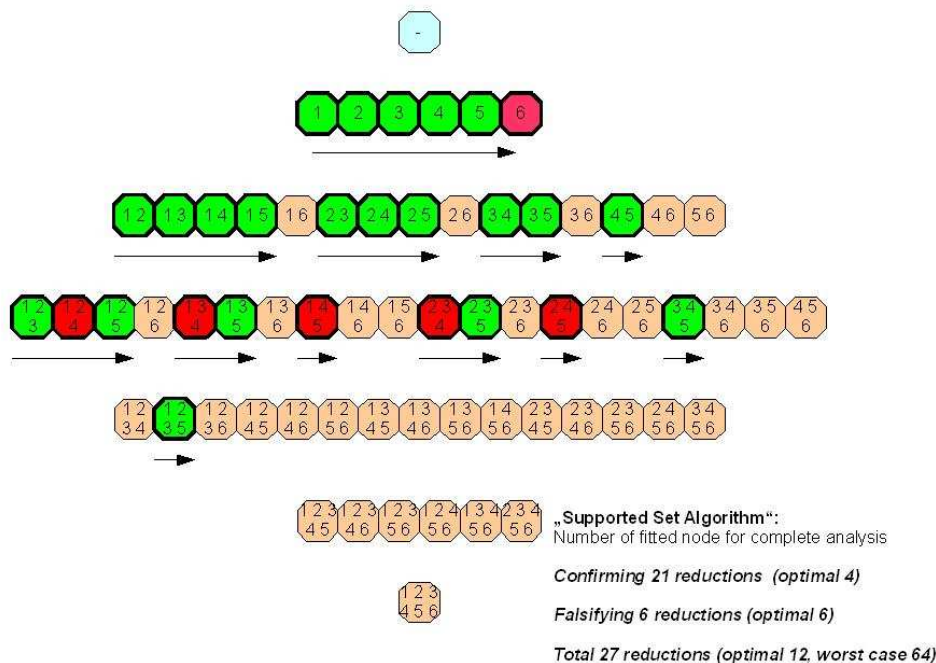


Figure 5.1.: Simulation of a supported set analysis of a reduction graph

The algorithm that can be designed in this way behaves identical to applying a standard breadth first search algorithm on the reduction graph. However we decided to introduce this approach from the item set mining perspective to better illustrate it's strengths and weaknesses.

A clear strength of this approach is that it will always find all maximal valid reductions while never considering any non-minimal invalid reductions. This is especially strong if the reduction graph contains a large ratio of non-minimal invalid reductions, for example in a model that allows us only to remove very few reactions. This algorithm will also provide fitted model for each reduction, which might be useful in further statistical analysis of the kinetic parameters in reduced submodels.

However this advantage comes at a price, since model fitting has to be run for each valid reduction (including all non-maximal valid reductions) the number of model fitting steps required can be quite large. This disadvantage should not be under estimated; since it may very well be possible for a model to have thousands or even millions of valid reductions the computational time required for this approach can easily become an inhibiting factor. However for smaller model with a moderate sized reduction graph or for the exploration of a partial solution of a larger model this approach seems a valid strategy.

5.2. Branch and bound algorithm

Another class of standard algorithms that is easily applied to discrete set based optimization problems are different variants of branch and bound algorithms. It should be noted that the classification of branch and bound for discrete algorithms isn't entirely consistent in the literature (e.g. sometimes cases that degenerate to simple depth/breadth first graph search are explicitly excluded); we base our classification on the definitions given by [Dieter Jungnickel *Graphs, Networks and Algorithms*, Springer]. Here branch and bound algorithms are simply defined as algorithms that split the remaining solutions in each step into two or more subsets that are then further explored using some heuristic.

This kind of algorithm can be applied to our reduction problem very easily. We start with all possible solutions in a single set. With each step we split the remaining solutions by assuming that some reaction isn't included in our reduction. This is analogous to moving to some node deeper in the reduction graph.

5.2.1. Last In First Out Branch & Bound

A very simple branch and bound strategy is to simply follow one branch of solutions until the remaining set of possible solutions doesn't contain any more valid solutions and then backtrack to the last split that has some unconsidered solutions, start a new branch by following this solutions until the entire space of solutions has been explored. This is efficiently implemented using some kind of stack memory to keep track of all splits. Unless further branch or cut criteria are employed this simple case is also identical to running a depth first graph search algorithm on the reduction graph data structure.

The performance of this strategy is largely dependent on chance; as we stated in 4 any complete solution will be required to attempt the fitting of all minimal valid and maximal invalid reductions. This can be considered the best case runtime. In an "ideal" run the branch that is followed will find all minimal invalid reductions without considering any non minimal invalid reductions. Once this has happened all branches will proceed directly towards the maximal valid solutions. However this event seems unlikely as in this simple approach no heuristic is employed to steer the current branch towards maximal invalid reductions. It's difficult to give any amortized runtime estimation for this algorithm, as both random factors and the model to be reduced influence the behavior of this algorithm in non trivial ways.

A main argument that could be made against this algorithm is that it doesn't take any information about the model to be reduced into consideration. This seems an ineffective strategy, as it implies that the model we try to reduce has no properties that influence how it can be reduced that can be analyzed without actually trying to reduce it. As our general opinion is that the analysis of kinetic parameters, topological properties and other features of the biochemical model should be taken into consideration this algorithm is more of a "baseline" for the design of more advanced branch and bound strategies.

5.3. Greedy search

Greedy algorithms are in most set based problems algorithms that will sort all possible elements according to some heuristic. The algorithm will then try, according to the order of this sorting, to include each item exactly one time. If an item is included when it is first considered it will also be included when every subsequent item is considered. The algorithm has now way to "trace back" later and decide to exclude a previously included item. If multiple greedy runs are to be performed for a given problem a common strategy is to add some kind of small random permutation to the order in which the elements are considered. This is done to introduce a non deterministic element to the greedy search as otherwise each subsequent search would be identical to the first greedy search.

The heuristics used for the ordering follow the same design ideas as those in section 5.2.2; advantageous strategies will usually involve the identification of small invalid or large valid reductions. The actual heuristics used are described in section 6.2.

6. Advanced reduction strategies

6.1. Boolean analysis of MA-network model

It has already been mentioned (for example in section 4.2.2) that one of the potential problems of the set based reduction approaches is that both obviously invalid and redundant solutions will be considered by the standard search algorithms employed unless we explicitly check for these cases. Examples for such cases are given in illustration 6.1. As considering a solution will involve multiple computationally expensive fitting attempts this should be avoided.

The examples in illustration 6.1 should give a good intuitive idea of the problematic cases which can be loosely characterized as disconnected outputs and constitutively inactive reactions. We will give a more formal definition of these two cases based on standard concepts of the analysis of nonlinear systems. This formal definition will then be used to design Boolean expressions that allow us to determine whether a reduction is valid. Finally we will show that reduced ordered binary decision diagrams allow us an implementation of a framework that supports efficient analysis of reductions based on these Boolean expressions.

6.1.1. Controllability and observability

We will adapt the concepts of controllable and observable states as defined in [19]. Consider a system of states where:

- Each state has a numerical value.
- The change of each state is given by an ODE dependent on any number of states in the system.
- A subset of states will be defined as input states.
- A subset of states will be defined as output states.

Our biochemical network model as defined in section 2.2 can be understood as a system with this kind of properties. Each state corresponds to a signaling molecule, the numerical value of the states denoting the concentration of the molecule. All states that have an initial concentration > 0 are considered input states, all states that can be experimentally measured are output states. For now we will ignore the case where a set of states is measured. In this framework we consider a state S_1 :

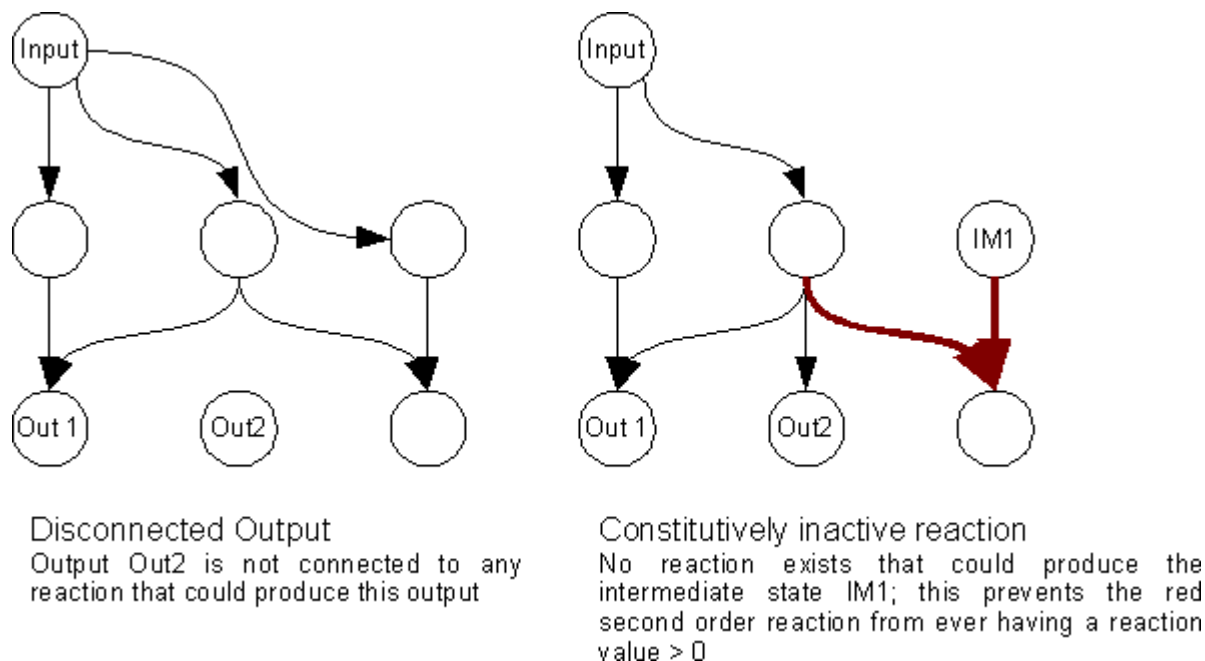


Figure 6.1.: Illustration of invalid and redundant reductions

- **controllable** by a state S_2 if for any two different starting values $S_2(t_0)_1$ and $S_2(t_0)_2$ it can be shown that $S_1(t||S_2(t_0)_1) \neq S_1(t||S_2(t_0)_2)$.
- **globally controllable** if it is controlled by any number of input states.
- **observable** if this state controls any output state.
- All input states are controllable by definition.

These are standard concepts in system analysis. We will now consider some special adaptations to our reduction approach. As all non input states have an initial value of zero any state not controlled by an input state will always have a concentration of zero. Therefore any uncontrollable state that is observable will be called **pseudo observable** as it could in theory be observed if it ever could have a concentration $\neq 0$.

So far these definitions are limited to the characterization of states. We will now extend them to reactions. Analogous to the state definitions a reaction R_1 is:

- **controllable** by a state S_1 if for any two different starting values $S_1(t_0)_1$ and $S_1(t_0)_2$ it can be shown that $R_1(t||S_2(t_0)_1) \neq R_1(t||S_2(t_0)_2)$. A reaction is considered **globally controllable** if it is controlled by any number of input states. A reaction can only be controllable if all its substrate and catalyzing states are also controllable, since we already established that any uncontrolled state will always have a concentration of zero. Therefore the product of the concentrations of all substrate and catalyzing states of the reaction will always be zero if at least one substrate or catalyzing state of the reaction is uncontrolled. This effect is analogous to the distinction between pseudo observable and non-pseudo observable states.

- We'll define *observability* of a reaction slightly different. A reaction is observable if it is controlled and any of its substrate- or product states is observable. This allows reactions to act as a sink - while the product of a reaction might not be measured, the fact that it consumes input might influence the output.

We will note the following conclusions about the interaction between states and reactions in regards to controllability and observability:

(Conclusion 6.0.1) If a reaction is controllable then all substrate and catalyzing states of the reaction have to be controllable. If any substrate state is uncontrollable then the reaction value will always be zero and the reaction itself would be uncontrollable.

(Conclusion 6.0.2) If the product states of an observable reaction are unobservable then at least one of its substrate states has to be observable.

(Conclusion 6.0.3) If a non-input state is controllable then at least one of the reactions producing this state is controllable.

(Conclusion 6.0.4) If a non-input state is observable then at least one controllable reactions that produces this state is also observable.

Some connection between these conclusions and the initially mentioned invalid and redundant reductions is easily established:

- A reduction is invalid if any output state of the network is uncontrollable.
- A reduction is redundant if the reduced model contains any unobserved reactions, as per definition any reduction that removes these unobserved reactions will behave identically.

It should be noted that these conditions for logical redundancy / invalidity are sufficient but not necessary. More sophisticated logical reasoning (such as not only considering whether all output states are controlled by any input states but also by which input states and whether this pattern of influence contradicts experimental evidence. We decided to leave this as a possibility for further works since most of these advanced logical approaches would have required significantly more implementation efforts then we deemed reasonable, considering that the main focus of this thesis isn't a purely logical analysis but an incorporation of different approaches. Practical application of these rules (see section 7.1) that the application of the rules established here are sufficient to reduce the reduction graph of a the CD95 model (20 states, 30 reactions) by several orders of magnitude.

6.1.2. Identification of invalid reduction

We will formally derive a recursive Boolean formula that helps us identify invalid reductions in the reduction space based on the model topology without any need for simulation. We will base this Boolean expression on one of the results from the prior section. We observed that it is a sufficient property for an invalid reduction if any output state of the model is uncontrollable. We will need some definitions for this expression:

- R_{total} is the set of all reactions in the model.

- S_{total} is the set of all species in the model.
- S_{input} is the set of all input states (states with an initial concentration > 0)
- S_{output} is the set of all output states (measured states)
- $S_{substrate}(i)$ is the set of all species that are substrates for reaction $r_i \in R_{total}$
- $S_{catalyze}(i)$ is the set of all species that are catalysts for reaction $r_i \in R_{total}$
- $S_{product}(i)$ is the set of all species that are products for reaction $r_i \in R_{total}$
- $R_{produces}(i)$ is the set of all reactions that state $s_i \in S_{total}$ is a substrate for.
- $R_{consumes}(i)$ is the set of all reactions that state $s_i \in S_{total}$ is a product of.
- $R_{controllable}$ is the set of all controllable reactions. Note that this set is unknown to us. We will utilize it in intermediate results but it may not be included in the final expression.
- $S_{controllable}$ is the set of all controllable states. Note that this set is unknown to us. We will utilize it in intermediate results but it may not be included in the final expression.
- $R_{observable}$ is the set of all reactions. Note that this set is unknown to us. We will utilize it in intermediate results but it may not be included in the final expression.
- $S_{observable}$ is the set of all observable states. Note that this set is unknown to us. We will utilize it in intermediate results but it may not be included in the final expression.

Note that all sets defined in this way result from the model topology and are known unless specifically stated otherwise. By definition every input state is controllable. Checking whether a state S_i is controllable is simple, as we know which states are input states. Formally this can be stated as $s_i \in S_{input}$. Conclusion 6.0.3 states that a necessary condition for a non-input state to be controllable is that at least one reaction producing this state has to be controllable. We can use this to formulate a necessary condition $c_{nec_contr_1}(s_i)$ for a state s_i to be controllable.

$$c_{nec_contr_1}(s_i) = (s_i \in S_{input}) \vee (\exists j : r_j \in R_{produces}(i) \wedge r_j \in R_{controllable}) \quad (6.1)$$

Since the set $R_{controllable}$ is only given by implicit definition but not known to us we will further relax our necessary condition by only demanding that r_j fulfills the necessary condition defined in conclusion 6.0.1. This condition demands that all catalyzing and substrate state of a reaction have to be controllable for the reaction to be controllable. We will call the necessary condition for reaction r_i to be controllable $c_{nec_contr_2}(r_i)$.

$$c_{nec_contr_2}(r_i) = (\forall s \in S_{substrate}(r_i) : s \in S_{controllable}) \quad (6.2)$$

$$\wedge (\forall s \in S_{catalyze}(r_i) : s \in S_{controllable})$$

$$c_{nec_contr_1}(s_i) = (s_i \in S_{input}) \vee (\exists j : r_j \in R_{produces}(i) \wedge c_{nec_contr_2}(r_j)) \quad (6.3)$$

Now we have to remove the dependency to $S_{controllable}$ as this set is only defined implicitly as well. We will do this again by substituting a necessary condition for the checks if $s \in S_{controllable}$. We will substitute this by $c_{nec_contr_1}(s)$

$$c_{nec_contr_2}(r_i) = (\forall s \in S_{substrate}(r_i) : c_{nec_contr_1}(s)) \quad (6.4)$$

$$\wedge (\forall s \in S_{catalyze}(r_i) : c_{nec_contr_1}(s))$$

This allows us to formulate a recursive definition for $c_{nec_contr_1}(s_i)$:

$$c_{nec_contr_1}(s_i) = (s_i \in S_{input}) \vee (\exists j : r_j \in R_{produces}(i) \wedge \quad (6.5)$$

$$(\forall s \in S_{substrate}(r_j) : c_{nec_contr_1}(s)) \wedge$$

$$(\forall s \in S_{catalyze}(r_j) : c_{nec_contr_1}(s))$$

Equation 6.5 is a necessary condition for all controllable states. However the implementation of this function in a logical framework has to be performed carefully as the necessary condition might result in a self reference. Consider the following example:

Example 6.1

States A, B, C

Outputstates : $\{ \}$

Reactions $AtoB, BtoC, CtoA$

$$\begin{aligned} c_{nec_contr_1}(C) &= (C \in \{ \}) \vee c_{nec_contr_1}(B) \\ &= (C \in \{ \}) \vee (B \in \{ \}) \vee c_{nec_contr_1}(A) \\ &= (C \in \{ \}) \vee (B \in \{ \}) \vee (A \in \{ \}) \vee c_{nec_contr_1}(C) \\ &= c_{nec_contr_1}(C) \vee \dots \end{aligned} \quad (6.6)$$

While this is obviously correct we gain no formal information. However if we consider the implications of such a situation informally we find that the logical implication for our system is:

In order for s_i to be controlled by any other state, s_i has to be controlled first

We will omit the formal proof for this, as it would be rather complicated without providing any real insight into the problem. The algorithmic implementation of this additional conclusion is rather simple, e.g. can be achieved by some kind of tracing variable that prevents performing a recursive call that has already been performed.

6.1.3. Identification of redundant reductions

We will base the identification of redundant reductions on the necessary condition that any model that contains unobserved reactions is redundant as derived in section 6.1. We will consider the following necessary but not sufficient conditions for a reaction r to be observable:

- Reactions that are observable have to be controllable.
- If a reaction is observable at least one of its substrate- or product states has to be observable.

We can use the results from section 6.1.2 to check the controllability of any state. This allows us to derive the necessary condition $c_{nec_obs_1}$:

$$c_{nec_obs_1}(r_j) = \forall s_i \in S_{substrate}(i) : c_{nec_contr_1}(s_i) \quad (6.7)$$

The second necessary condition is that at least one of the substrate- or product states of a reaction has to be observable for the reaction to be observable. We will call this $c_{nec_obs_1}$:

$$c_{nec_obs_1}(r_j) = \begin{aligned} &\exists s \in S_{substrate}(j) : s \in S_{observable}(i) \vee \\ &\exists s \in S_{product}(j) : s \in S_{observable}(i) \end{aligned} \quad (6.8)$$

Utilizing this necessary condition requires us to consider the observability of a state. A necessary condition for an observable state is that it either is an output state or the substrate of an observable reaction. We will use this to define a necessary condition for observable states $c_{nec_obs_2}$.

$$c_{nec_obs_2}(s_i) = \begin{aligned} &s_i \in S_{output} \vee \\ &\exists r_j \in R_{produces}(i) : r_j \in R_{observable} \vee \\ &\exists r_j \in R_{consumes}(i) : r_j \in R_{observable} \end{aligned} \quad (6.9)$$

Now we proceed analogously to section 6.1.2, finding that:

$$\begin{aligned} c_{nec_obs_2}(s_i) &= s_i \in S_{output} \vee & (6.10) \\ &\exists r_j \in R_{produces}(i) : r_j \in R_{observable} \vee \\ &\exists r_j \in R_{consumes}(i) : r_j \in R_{observable} \end{aligned}$$

$$\begin{aligned} c_{nec_obs_2}(s_i) &= s_i \in S_{output} \vee & (6.11) \\ &\exists r_j \in R_{produces}(i) : c_{nec_obs_1}(r_j) \vee \\ &\exists r_j \in R_{consumes}(i) : c_{nec_obs_1}(r_j) \end{aligned}$$

$$\begin{aligned} c_{nec_obs_1}(r_j) &= (\exists s_i \in S_{substrate}(j) : c_{nec_obs_2}(s_i)) \vee & (6.12) \\ &(\exists s_i \in S_{substrate}(j) : c_{nec_obs_2}(s_i)) \end{aligned}$$

$$\begin{aligned} c_{nec_obs_1}(r_j) &= (\exists s_i \in S_{substrate}(j) : s_i \in S_{output} \vee & (6.13) \\ &\exists r_l \in R_{produces}(i) : c_{nec_obs_1}(r_l) \vee \\ &\exists r_l \in R_{consumes}(i) : c_{nec_obs_1}(r_l)) \vee \\ &(\exists s_i \in S_{product}(j) : s_i \in S_{output} \vee \\ &\exists r_l \in R_{produces}(i) : c_{nec_obs_1}(r_l) \vee \\ &\exists r_l \in R_{consumes}(i) : c_{nec_obs_1}(r_l)) \end{aligned} \tag{6.14}$$

6.1.4. Implementation of Boolean model checking using BDD's

The implementation of the Boolean model checking has to be performed with care. The Boolean functions defined in section 6.1.3 and 6.1.2 describe a space of $2^{\|R_{ele}\|}$ different logical assignments. This will prohibit the application of explicit boolean representations like truth tables. Binary decision diagrams are a datastructure that is often utilized for this kind of implicit representation. We will limit us to a small review of the properties of BDD's and refer to other sources, such as [20] for more detailed informations.

BDD's are graph based representations of Boolean equations. They provide implementations for all Boolean standard operations and can be shown to be functionally identical to other Boolean structures. A BDD allows us to check in constant time whether the function it encodes is satisfiable. This is no contradiction to the classification of $SAT \in NP$, as in the worst case the construction of a BDD requires exponential time complexity. The average complexity for the construction of a BDD depends on the exact nature of the Boolean equation. Boolean equations with high degrees of redundancy can often be encoded in significantly better than exponential time. A main strength of BDD's are efficient heuristics for the application of 'AND' and 'OR' operations. This properties have been the basis for prior analysis of biological models with BDDs by [21].

Basing our implementation strategy on the work of Garg et.al. and implementing BDDs representing the Boolean expressions defined in 6.1.2 and 6.1.3 we found that we could

utilize BDDs to efficiently check the validity and redundancy of models with a reduction space $> 10^{10}$ possible states.

6.2. Heuristic strategies of priority estimation

We already discussed the importance of heuristic strategies to increase our probability of evaluating reductions that provide us with a "better than random" information gain in chapter 5. Two distinct basic strategies exist; strategies that focus on the structuring of the reduction space and strategies that focus on model intrinsic properties. We implemented several reduction heuristics. We tried to compare the performance of these strategies but failed due to the already mentioned insufficient supply of unreduced models. Attempts to benchmark the different heuristics on artificial models were deemed insufficient as the results were determined by the algorithm employed in the generation of the artificial models. Most reduction strategies could be implemented in different ways that change several aspects of their behavior. As an example kinetic parameter based analysis might use the average, minimal or maximal value that a parameter has in different fitting runs. Again, detailing these small differences without any basis for comparing their quality seems uninteresting. We will limit us therefore to a general introduction of the ideas of the different reduction heuristics omit a detailed description of all implemented heuristics.

6.2.1. Maximum information gain strategy

Based on the rules for automated validation and invalidation in chapter 4.2.2 we can define a possible strategy that focuses on exploiting the structured nature of the reduction graph. We will use some basic definitions:

- R_{inv} is the set of all invalid reductions
- R_{val} is the set of all validated reductions
- R_{red} is the set of all redundant reductions
- $R_{anc}(i)$ is the set of all ancestors of reduction i
- $R_{desc}(i)$ is the set of all descendant of reduction i

Let us now consider how many reductions are validated/invalidated by evaluating a reduction. Validating a reduction validates all its ancestors. However we gain no new information regarding reductions that have already been validated. We also assume that redundant reductions will not provide any information gain as we expect that the reduction they are identical to will be considered in the set of ancestors. Thus we validate all ancestors of the reduction considered that have not yet been considered or are redundant. This can be formalized as:

$$ig_{val}(i) = \|R_{anc}(i) \setminus (R_{val} \cup R_{red})\|$$

We can estimate the number of states we invalidate when evaluating a single reduction as invalid in a symmetric estimation:

$$ig_{inv}(i) = \|R_{desc}(i) \setminus (R_{inv} \cup R_{red})\|$$

Now we can define the estimated information gain ig_{est} when evaluating a reduction:

$$ig_{est}(r_i) = p_{isValid}(r_i) * ig_{val}(i) + (1 - p_{isValid}(r_i)) * ig_{inv}(i)$$

The core of this heuristic is the estimation of $p_{isValid}(r_i)$. Some strategies to estimate $p_{isValid}(r_i)$ are:

- Assume that $p_{isValid}$ is a constant value, e.g. $p_{isValid} = 0.5$
- Estimate $p_{isValid}$ based on the fraction of all evaluated states that are valid, e.g. $p_{isValid} = \|R_{val}\|/\|(R_{val} \cup R_{inc})\|$
- Estimate $p_{isValid}(r_i)$ based on the fraction of all evaluated states with the same number of elementary reductions as r_i that are valid

6.2.2. Analysis of prior fitted kinetic parameters

The basic idea of this strategy is to derive a score based on a reductions already fitted ancestors. This can be done in a number of ways. An example might be to consider the average kinetic parameters of all reactions that are present in the ancestors but not in the current reduction. A variant of this might limit the ancestors only to the parents or the closest ancestors of a reduction. The larger the average kinetic parameter of a reduction to be removed is, the more likely it for the reduction to influence the behavior of the model. Depending on whether we desire try to find maximal valid or minimal invalid reductions we can order all candidate reduction in ascending or descending order of the average kinetic parameter of the removed reduction.

7. Application of the reduction algorithm

One major concern was finding a suitable way to demonstrate and analyze our reduction framework. We already stated the problem of the lack of available unreduced networks in integrated databases. This prohibited a large scale comparison of different reduction strategies. As no standard approach for randomizing biochemical models while retaining characteristic parameters could be found this essentially prohibited a quantitative comparison of our different heuristics.

Instead and in agreement with our scientific supervisor Fabian Theiss we decided to focus on a proof of principle analysis. We decided that a reduction of the CD95 model or a medium sized artificial model would suffice to illustrate the capabilities of our framework. A complete reduction of the CD95 model failed due to computational limitations; a medium sized artificial model could be fitted completely. In this chapter we will detail both the limitations to the reduction of the CD95 model and the reduction of the artificial model.

7.1. Analysis of the CD95 model

The initial CD95 model contains 34 reactions and 25 different states. If we consider all combinatorial possibilities to remove any subset of 34 edges we find that roughly seventeen billion different possibilities exist. Any analysis of a dataspace of this proportions has to be performed with care.

Several technical challenges had to be overcome in order for our analysis algorithm to handle the complete CD95 model. Explicit data structure could not be employed to save intermediate results. Instead an indirect reductionspace representation had to be designed, mapping reduction ids to binary values. Multiple BDDs were utilized to keep track of validated, redundant and unknown reductions. To test the stability of our implementation a limited analysis was performed on the CD95 model. The analysis included the initial analysis of obviously invalid and redundant states, the fitting of the unreduced model and 50 fitting steps using a greedy search algorithm. This took roughly 50CPU hours parallelized on 24 cores. Further parallelization could have reduced this time at most by 50%. The results of this partial fitting are summarized in table 7.1

We found that our framework was able to handle the size of the reductionspace in this partial reduction. We could confirm that the implicit representation functions as designed since more than 10^{10} states could be invalidated. The greatest information gain

<i>Analysis performed</i>	<i>Time required for analysis</i>	<i>Unknown reductions</i>
Initial situation	-	$\approx 1.71 * 10^{10}$
Analysis of invalid states	<2 minutes	$\approx 2.73 * 10^8$
Analysis of redundant states	<2 minutes	$\approx 2.78 * 10^7$
Analysis of 50 steps greedy run	≈ 50 hours	$\approx 3.9 * 10^6$

Table 7.1.: Results: simulation of the CD95 model. Technical reasons made it impossible to measure the time for validity and redundancy analysis separately

was achieved by the exclusion of all invalid reductions (invalidating over 98% of the initial reductionspace). Removing redundant reductions removed almost another 90% of the reduction space. Without any parameterfitting 99.83% of the reduction space could be invalidated based on logical reasoning. We consider this the confirmation that using a setbased reduction strategy to structure the reduction graph is indeed a valid approach. The problem that a purely setbased structurization of the reductions results in a reductionspace that contains for the most part invalid reductions can be countered by implicit Boolean model checking based on the definitions of the elementary reductions.

The greedy run performed reduced the number of unknown states by further 89.8%, resulting in the evaluation of 99.97% of the original reduction space. However we also have to consider that the remaining reduction space after 50 reduction steps is still to large to completely enumerate. Considering that a model took on average one hour to fit the evaluation of every remaining reduction by individual fitting would require $3.9 * 10^6$ hours. This is a worst case scenario ignoring all validation and invalidation rules. We consider the small sample size of 50 reduction steps insufficient to predict the further performance of the greedy reduction strategy. Some factors might favor "early" fitting run; for example the identification of an essential reduction that is required for fitting might reduce the remaining reduction space by up to 50%. As a "best case" scenario we could assume that the current information gain per reduction step ration remains constant, e.g. 90% of the reduction space are evaluated every 50 reduction steps. This would imply that only about 350 more hours of analysis would suffice to completely analyze the reduction space. Based on our lack of further data we can not estimate to which extreme the real further time requirements tend. It is planed to explore the continued reduction of the CD95 in a follow up project.

7.2. Analysis of a medium sized toy model

We decided to analyze the reduction of a smaller artificial model that was created in a way that allowed easier parameter fitting. The goal of this second analysis was not an estimation of the time requirements of different reduction strategies as we consider the performance of our heuristics on artificial models to be possibly inconsistent with their behavior on real models. The goal was rather to illustrate the possible analyzes that could be gained from a complete reduction of a model.

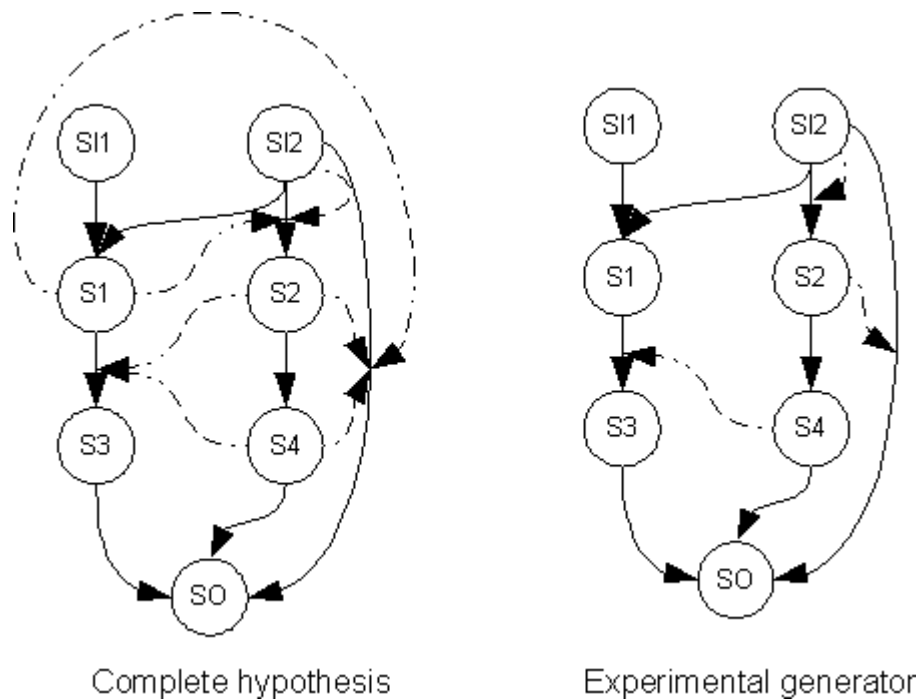


Figure 7.1.: Complete artificial model and the submodel used to generate experimental data

7.2.1. Model used

The artificial model was designed to resemble the DISC-dimer subnet of the CD95 model. We used the same kinetics and focused on the presence of alternative catalyzing species for different reductions. Some alterations had to be made to introduce asymmetry into the model that in the CD95 model was introduced upstream of the subnet. A subset of all reactions of the complete artificial model was used to simulate an experiment using two different initial conditions. All kinetic parameters were randomized for this simulation. The complete artificial model and the subgraph used to generate experimental data are illustrated in figure 7.1. The input states were state SI1 and SI2, the only measured output was SO

7.2.2. Methods

We reduced the model using a priority branch and bound strategy. Initial validity and redundancy analysis was performed as described in chapter 6.1. We performed two complete reductions, one using an information gain based strategy the other using a kinetic parameter based strategy.

7.2.3. Results & Discussion

After initial Boolean analysis the reduction space contained about 700 reductions of unknown status. Both strategies required about 40 reduction steps to achieve a complete reduction of the toy model. The complete reduction graph is illustrated in figure 7.2. We

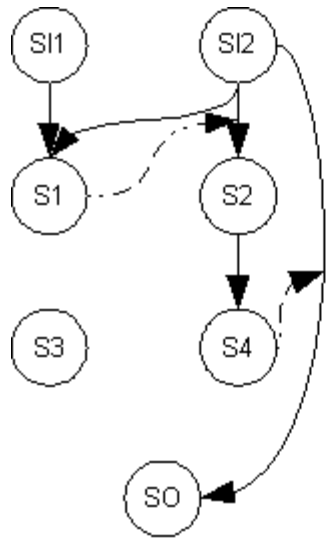
can see that the number of valid reductions decreases as we remove more reductions. We can also recognize the way validity is promoted upward from valid descendants. Different reduction "paths" can be recognized that get thinner deeper in the reduction graph and "merge" in upper regions. A motivation for further analysis might be the exploration of the branching of these paths. Identifying points where different paths merge might help in the design of experimental setups that differentiate between different possible reductions. Invalidating a reduction shortly after a branching invalidates most of the branch because of the descendant-invalidation rule. If we want to design an experiment that invalidates one of two possible reduction paths we could try to identify reductions that are one hierarchy level below the branching point. The reductions identified in this way could be simulated for different initial conditions that are reproduceable in an experimental setup. If any of these candidates for experimental setups produces data that is supported only by reductions in one of the two branches we expect the experimental data we can generate in this way to invalidate at least one of the two branches.

Based on a detailed analysis of the reduction graph we can identify four different maximal reductions, as illustrated in figure 7.3. All of these maximal reductions are sparse; several reactions are not included in any of the minimal solutions. Notably is the absence of the production of output using either state S3 or S4. In fact none of the maximal reductions produces S3 at all. This could suggest that state S3 is a possible candidate for a state that does not influence the modeled system. A possible follow up experiment could be designed around a S3 knockout to either confirm or reject this assumption.

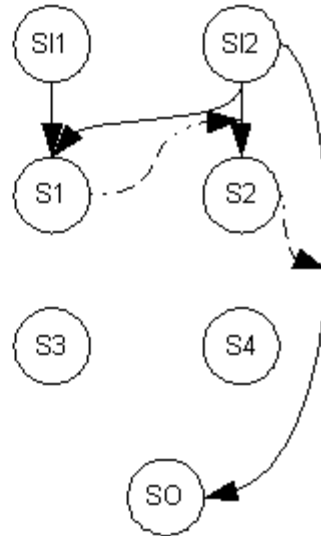
Other reactions are common to almost all maximal reductions. Every maximal reduction contains a reaction that converts SI2 to S0. This reaction seems to be a key element of the systems behavior. Initially we proposed three different possible catalysts for this reaction: S1, S2 and S4. We find that S1 is not included in any of the reductions. This implies that the design of further experiments should focus on distinguishing between the possible catalysts S2 and S4.



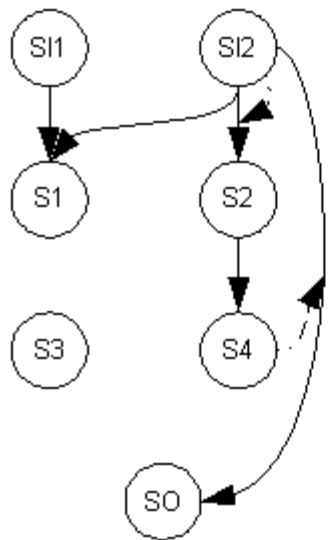
Figure 7.2.: Complete reduction graph of the artificial model employed; yellow states are valid, violet states are invalid.



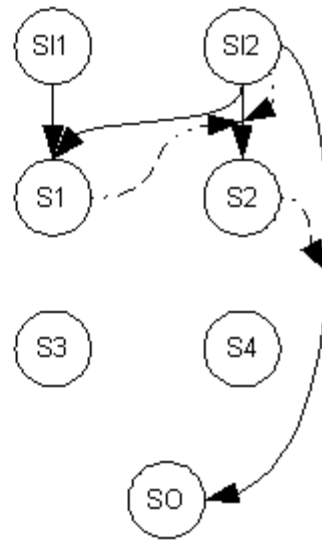
Minimal solution 1



Minimal solution 2



Minimal solution 3



Minimal solution 4

Figure 7.3.: Illustration of all minimal reductions

8. Discussion of results and further perspectives

When comparing our results to our initial motivation we found that we could not achieve our initial goal of running a complete reduction analysis of the CD95 model. This was mainly caused by computational limitations, e.g. the lacking possibility of massively parallelized fitting. The CD95 model required a lot more time for fitting than we initially expected. Consultation with Nicolai Fricker, the original designer of the model we used confirmed that he could neither supply us with reasonable constraints for the kinetic parameters nor with a fast reliable fitting implementation. We consider this problem during the analysis of the CD95 model acceptable. The problems we encountered were centered around the implementation of efficient modelfitting which wasn't the focus of our project.

To validate our general idea we performed automated model reduction on a number of artificial models designed using kinetics that resembled the CD95 model on a smaller scale. We presented the analysis of a smaller artificial utilizing the limited resources available. The result of our analysis helped to illustrate possible applications on real models by providing data that allowed us to identify "key reactions" in the artificial model that dominated the models behavior. We performed the reduction using two different heuristics, a maximum information gain- and a prior parameter fitting analysis based heuristic, in conjunction with a priority based branch and bound algorithm and found identical results. The limited number of real models available for testing did not allow benchmarking different heuristic strategies within the time frame of this project.

While the complete fitting of the CD95 model failed we were able to run a limited fitting run with 50 attempted reductions. Although no significant insight could be gained from such a limited analysis regarding the complete reductionspace of the CD95 model we could demonstrate the ability of our framework to handle a model with a reduction graph of 2^{34} possible elements. This demonstrates the power of our strategy to represent the reduction graph in an implicit manner using binary decision diagrams as a supporting data structure. It also demonstrates our ability to identify a significant number of redundant and invalid states in the reduction graph of the CD95 model, reducing the number of possible reductions by over 98%.

We also overcame a number of technical obstacles that were related to the fitting of models. The framework we implemented supports parallelized fitting using a simulated annealing algorithm in different variants. Utilizing the matlab parallelization toolbox on a multi core machine is the easiest variant and does not require the user to be experienced in the compilation of matlab standalone application. Using the matlab compiler we were also able to produce stand alone binaries that allow parallelized fitting on broader variants

of cluster architectures. This was utilized using a batch system based on the sun grid engine to parallelize our fitting runs on a cluster of 24 cores.

Based on our frameworks ability to deal with large reduction spaces and the ability to adapt the parallelized fitting to different cluster architectures we decided that the analysis of the CD95 model using our framework should be continued beyond this project. It is currently planned to adapt our framework to larger clusters and perform an analysis in the order of 1000 cpu hours.

Although no concrete projects are planned yet we consider further applications for our basic framework. A possible application could be the design of optimized experimental setups. A rough outline for this application could be:

- Start by reducing the initial model using our framework.
- Identify "important" reductions in the reduction graph. Examples for important reductions might be all maximal valid reductions as well "branching" reductions, e.g. reductions that are ancestors of multiple maximal valid reductions while none of their children are ancestors to all the same maximal valid reductions.
- Cluster all important reductions by the similarity of their possible parametrization
- Try to find simulations that result in different behavior in different clusters.
- The simulations that distinguish optimally between different clusters are the basis for further experimental design.

Considering the prospect of possible further applications we think our project succeeded in demonstrating the possible power and flexibility of a structured reduction framework.

Bibliography

- [1] Report on EU–USA Workshop: How Systems Biology Can Advance Cancer Research, R. Aebersold et.al. ,Molecular Oncology 2009
- [2] Why We Need Quantitative Dynamic Models, Ravi Iyengar, Sci. Signal., 31 March 2009
- [3] N. van Riel. Dynamic modeling and analysis of biochemical networks: mechanism-based models and model-based experiments. Brief. Bioinform. (2007)
- [4] P.J. Ingram, et al. Network motifs: structure does not determine function. BMC Genomics. (2006)
- [5] A structured approach for the engineering of biochemical network models, illustrated for signalling pathways, R. Breitling et.al., Brief Bioinform. 2008
- [6] M. E. Peter and P. H. Krammer, The CD95(APO-1/Fas) DISC and beyond. Cell Death Differ 10 2003
- [7] I. Lavrik I et. al., Death receptor signaling, J Cell Science 2005
- [8] Briggs, G. E., and Haldane, J. B. (1925) A Note on the Kinetics of Enzyme Action, Biochem J 19
- [9] Erdős, P.; Rényi, A. (1960). "The Evolution of Random Graphs".
- [10] Modeling and Simulation of Genetic Regulatory Systems: A Literature Review,H. de Jong, J Comput Biol. 2002;9(1):67-103
- [11] Rakesh Agrawal, Ramakrishnan Srikant: Fast Algorithms for Mining Association Rules. In: VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA 1994
- [12] Milo et al., Network Motifs, Science (2002)
- [13] H. Schmidt, M. Jirstrand: Systems Biology Toolbox for MATLAB: A computational platform for research in Systems Biology, Bioinformatics Advance Access, November 2005
- [14] Tzafriri, A. R., and Edelman, E. R. (2005). On the validity of the quasi-steady-state approximation of bimolecular reactions in solution. J. Theor. Biol. 233, 343-350
- [15] <http://biocyc.org/>
- [16] <http://www.reactome.org/>

- [17] Exact model reduction of combinatorial reaction networks, H. Conzelmann et.al., BMC Systems Biology 2008
- [18] H. Schmidt, M. Jirstrand: Systems Biology Toolbox for MATLAB: A computational platform for research in Systems Biology, Bioinformatics Advance Access, November 2005
- [19] Isidori A., Nonlinear Control Systems
- [20] Henrik Reif Andersen. An Introduction to Binary Decision Diagrams. Lecture notes, available at <http://www.itu.dk/people/hra/notes-index.html>
- [21] Abhishek Garg et.al., An Efficient Method for Dynamic Analysis of Gene Regulatory Networks and in silico Gene Perturbation Experiments, 11th Annual International Conference, RECOMB 2007
- [22] JavaBDD: Implementation and documentation available at: <http://javabdd.sourceforge.net/index.html>
- [23] BuDDy: Implementation and documentation available at: <http://sourceforge.net/projects/buddy>

List of Figures

2.1.	Network representation of the CD95 model. It should be noted that the reaction that cleaves C3 and produces active C3 is potentially regulated by any C8 dimer, p43 dimer or p18; these regulatory influences could not be shown for reasons of illustrative clarity.	14
2.2.	Artificial model used to estimate the effect of initial condition fitting vs. kinetic parameter fitting	18
2.3.	Comparison of kinetic parameter and initial condition fitting	19
2.4.	Example for a possible reduction of a subnet of the CD95 model	21
4.1.	Illustration of the concepts of redundancy and obvious invalidity	32
4.2.	Illustration of the different topological relations states can have and of possible problems that can result from merging states	36
4.3.	Illustration of validation and invalidation in the reduction graph	40
5.1.	Simulation of a supported set analysis of a reduction graph	42
5.2.	Simulation of a Last In First Out branch and bound analysis of a reduction graph	44
5.3.	Simulation of a single run greedy analysis analysis of a reduction graph	45
6.1.	Illustration of invalid and redundant reductions	48
7.1.	Complete artificial model and the submodel used to generate experimental data	58
7.2.	Complete reduction graph of the artificial model employed; yellow states are valid, violet states are invalid.	60
7.3.	Illustration of all minimal reductions	61

A. External tools, packages and librarys used

We utilized the matlab statistics and bioinformatics toolbox for various calculations. The parallelization toolbox and the matlab compiler were required for parallelized fitting. All of these toolboxes are available as part of the matlab installation at the CMB.

We used the external matlab toolbox sbtoolbox [18] as a basic framework to solve the ODEs of mathematical models. Fabian Theiss provided us with a matlab implementation of an simulated annealing algorithm used for model fitting.

We utilized the Java interface JavaBDD [22] for all Boolean analyses. This interface is based on the C++ library BuDDy [23].