Chapter 3
Systematic Complexity Reduction of Signaling Models and Application to a CD95 Signaling Model for Apoptosis

Dennis Rickert, Nicolai Fricker, Inna N. Lavrik, and Fabian J. Theis

Abstract A major problem when designing mathematical models of biochemical processes to analyze and explain experimental data is choosing the correct degree of model complexity. A common approach to solve this problem is top-down: Initially, complete models including all possible reactions are generated; they are then iteratively reduced to a more manageable size. The reactions to be simplified at each step are often chosen manually since exploration of the full search space seems unfeasible. While such a strategy is sufficient to identify a single, clearly structured reduction of the model, it discards additional information such as whether some model features are essential. In this chapter, we introduce alternate set-based strategies to model reduction that can be employed to exhaustively analyze the complete reduction space of a biochemical model instead of only identifying a single valid reduction.

D. Rickert • F.J. Theis (✉)
Institute of Bioinformatics and Systems Biology, Helmholtz Zentrum München, Ingolstädter Landstraße 1, 85764 Neuherberg, Germany
e-mail: fabian.theis@helmholtz-muenchen.de

N. Fricker
Institute of Bioinformatics and Systems Biology, Helmholtz Zentrum München, Ingolstädter Landstraße 1, 85764 Neuherberg, Germany

Division of Immunogenetics, German Cancer Research Center (DKFZ), Heidelberg 69120, Germany

Bioquant, Heidelberg 69120, Germany

I.N. Lavrik
Department of Translational Inflammation Research, Institute of Experimental Internal Medicine, Otto von Guericke University, Leipziger Str. 44, 39120 Magdeburg, Germany
e-mail: inna.lavrik@med.ovgu.de

I.N. Lavrik (ed.), *Systems Biology of Apoptosis*, DOI 10.1007/978-1-4614-4009-3_3,

3.1 Introduction

A major problem when designing mathematical models of biochemical processes to analyze and explain experimental data is choosing the correct degree of model complexity. Minimalistic models that include only the core reactions of a regulatory pathway will often fail to capture all mechanisms and be unable to reproduce the experimentally observed dynamics. In contrast, models that include all possible interactions may suffer from overfitting, strongly diminishing their predictive and analytical value. This becomes especially severe in pathways where interacting molecules are modified by, or bind with multiple interaction partners, as is common in inter- and intracellular signaling.

A common approach to solve this problem is top-down: Initially, complete models including all possible reactions are generated; they are then iteratively reduced to a more manageable size. The reactions to be simplified at each step are often chosen manually since exploration of the full search space seems unfeasible. While such a strategy is often sufficient if the goal is limited to finding a single, clearly structured reduction of the model, other questions that could be of interest from a modeling point of view are not considered. Examples for such questions are:

- Are some model features essential, i.e., can never be reduced?
- What are the smallest versions of the model that are still viable?
- Is there a logical pattern common to all valid model reductions?

These questions will often require the analysis of all possible model reductions, a task that is both time consuming and repetitive, making it ill-suited to manual analysis. Explicit enumeration and testing of all possible reductions is prohibitively expensive, imposing the need to utilize heuristic search strategies. In this chapter we will discuss a set-based strategy that is suited to answer the questions posed above. While we explain the strategy in the context of ODE modeling with mass action kinetics, it can be adapted to a wide range of models, including SDEs, Boolean models, and stochastic simulations. We finish the chapter with an application to CD95 signaling.

3.2 Graphical Structuring of ODE Models

It is often helpful to organize ODE models into a graph-like data structure. Not only does this allow easy visualization of the model, it also allows the utilization of established graph-based operations, like the removal of edges, and graph-based algorithms, e.g., connectivity analysis with little adaption. This is frequently done by researches, however, depending on the exact application, the details of the representation differ. To avoid confusion, we give a short overview of the approach we utilize. An example for a mass action ODE that is represented by a graph is illustrated in Fig. 3.1. Note that while we limit ourselves to mass action kinetics in this chapter to avoid excessively complex notation, the general approach is also valid for more complex reaction kinetics, such as Michaelis–Menten kinetics.

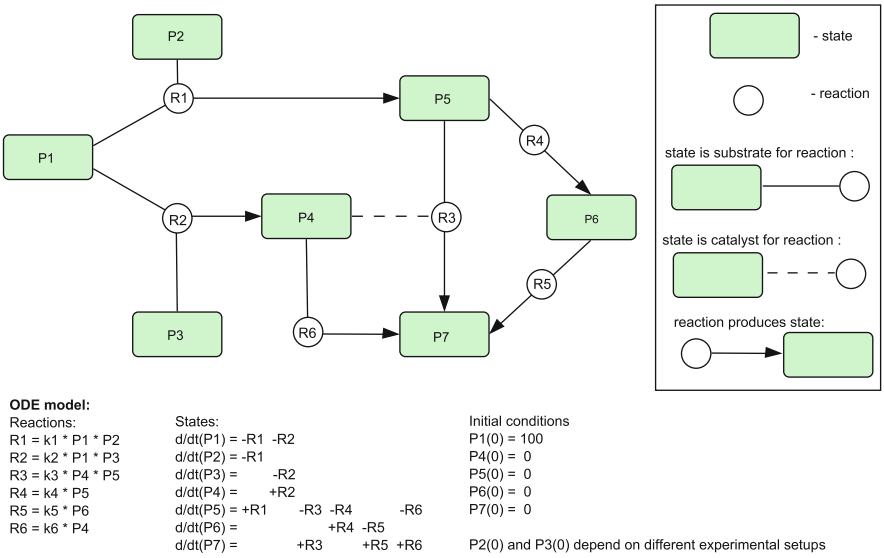

Fig. 3.1 A graphically represented ODE model. An ODE model represented as a bipartite graph. *Green boxes* are states (usually biochemical molecules), *white circles* are reactions. A state can be either a substrate (*normal line*) or a catalyst (*dashed line*) for a reaction. A directed line from a reaction to a state indicates that the reaction produces this state. In this model, the state P1–P3 act as input (e.g., signaling molecules, primary metabolites...), the state P7 is the final model output. P1 dimerizes with either P2 or P3. Both the P1–P2 dimer (P5) and the P1–P3 dimer (P4) can be activated independently into P7 (P5 via the intermediate P6, P4 directly). In addition, there is a cooperative mode of activation where P4 catalyzes the activation of P5

We focus on ODE models that can be decomposed into two types of elements: *states* and *reactions*. States represent the molecular species of the system we want to analyze, whereas reactions are the biochemical reactions that change one or more molecule into other products. For a given ODE model, we will usually know the value of all reactions depending on the current state values and the first derivate of all states depending on the current reaction values. This is sufficient to simulate the model using a numerical solver.

Based on the following definitions:

$S_{1...k}$	The set of all states
$R_{1...j}$	The set of all reactions
k_n	The kinetic rate of reaction n.
$\text{Subs}_R(R_n)$	is the index set of all states that are substrate of reaction n.
$\text{Subs}_S(S_n)$	is the index set of all reactions that consume state n.
$\text{Cat}_R(R_n)$	is the index set of all states that are catalysts of reaction n.
$\text{Cat}_S(S_n)$	is the index set of all reactions for which state n is a catalyst.
$\text{Prod}(S_n)$	is the index set of all reactions that produce state n.
$\text{stoich}(S_i, R_n)$	is the stoichiometric constant of state i in reaction n (the stoichiometric constant denotes how many molecules of each type participate in a reaction, e.g., for a homodimerization the stoichiometric constant is 2).

The value of reaction n at time t:

$$R_n = k_n \prod_{i \in (\text{Subs}_R(R_n) \cup \text{Cat}(R_n))} S_i^{\text{stoich}(S_i, R_n)}$$

The change of state m at time t:

$$\frac{\mathrm{d}}{\mathrm{d}t}(S_n) = + \sum_{i \in \text{Prod}(S_n)} \text{stoich}(S_n, R_i) * R_i - \sum_{i \in \text{Subs}_S(S_n)} \text{stoich}(S_n, R_i) * R_i$$

To generate the model graph for a given ODE:

For a graphical representation of this an ODE system, both states and reactions are considered nodes in a model graph. The model graph is bipartite graph, i.e., there will only be connections between reactions and states, but not between states and states or reactions and reactions.

- There is a directed edge from every state n to every reaction m if $n \in \text{Subs}_R(R_m)$, i.e., if state n is substrate for reaction. We label all these edges with stoich (S_n, R_m) and color them as substrate influences.
- There is a directed edge from every state n to every reaction m if $n \in \text{Cat}(R_m)$, i.e., if state n catalyzes reaction m. We label all these edges with stoich(S_n, R_m) and color them as catalytic influences.
- There is a directed edge from each reaction m to each state n if $m \in \text{Prod}(S_n)$, i.e., if reaction m produces state n. We label all these edges with stoich(S_n, R_m) and color them as productions.

It is easy to see that additional, more complex kinetics can be supported by simply increasing the number of different colors used to color the edges from states to reaction nodes.

3.2.1 Experimental Noise Model and Error Function

One of the most important tasks in model-driven systems biology is to evaluate how well a model explains experimental observation. Typically, we will want to run simulations of different experimental conditions, such as different intensities of stimulation and compare the resulting model dynamics to experimental measurements. Figure 3.2 illustrates how the simulation of biochemical models produces time course data for different experimental conditions.

While generating the experimental data is usually straightforward and can be done using established toolboxes such as the SBtoolbox 2 for Matlab (Schmidt 2006), the comparison is less straightforward and can be influenced by personal bias. To minimize this subjectivity, it is useful to quantify the difference between model and experimental data. This is often done by utilizing *error functions*. An error function quantifies the difference between time course data

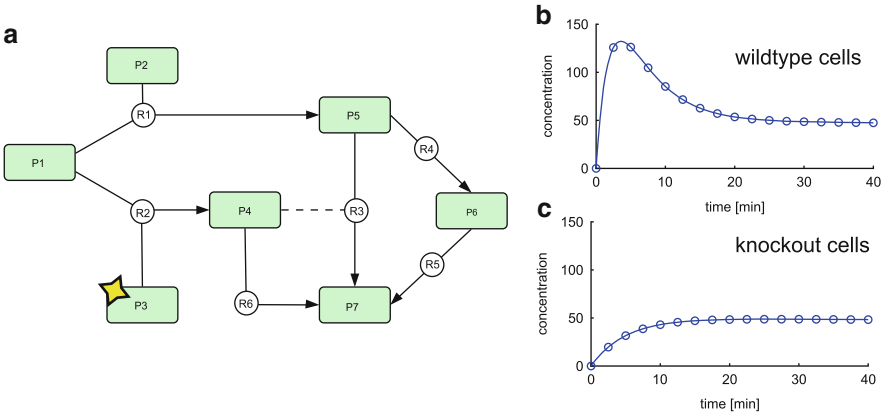

Fig. 3.2 The behavior of dynamic systems can be changed by different experimental conditions. (**a**) A typical experimental setup will often involve the reduction or complete knockout of one or more states of a dynamical system. In our example, the protein P3 is completely knocked out (set to concentration zero). (**b**) In wild-type cells, the reaction to stimulation is an early peak in activity (at around 5 min), followed by a slow decay to a constant concentration level of about 50. (**c**) In knockout cells the early peak at 5 min is missing. The later activation remains unaffected by the knockout

generated by the model and experimental observations. However both the generation and the interpretation of error functions and their values are nontrivial.

A major problem in systems biology is that data will often be very noisy; even repeated measurements of the same system at the same time can vary by 10% or more. This is caused both by measurement errors and by the high variability of biological systems. In contrast deterministic models such as ODEs will always reproduce identical results without any noise; therefore, it is unrealistic to expect an ODE to reproduce experimental observations perfectly.

To deal with this situation, we try to capture the experimental variance in an error noise model. For many different experimental setups, we observe that measurement errors and experimental variance are normally distributed $\boldsymbol{N}(\mu, \sigma^2)$ with a mean μ of zero and a variance σ^2 that depends on the exact experimental setup. We assume that the "basic" behavior of the system is determined by the deterministic model and that the difference between deterministic model and observed data is caused by the experimental variance. This is illustrated in Fig. 3.3.

$$\begin{aligned}\text{Observation} = {} & \text{deterministic behavior (ODE model)} \\ & + \text{model variance (stochastic component)}\end{aligned}$$

Based on this assumption we can calculate how probable it is to observe our experimental data. Initially we will only consider the case where a single variable (e.g., protein concentration) is observed at the time points $1...n$. This is mainly done to avoid confusing notation, in the end we will derive the total error value by simply summing over all individual error values. Let $x^{\text{obs}} = x^{\text{obs}}_{1...n}$ be the vector of our experimental observations if only a single replicate of the experiment is performed.

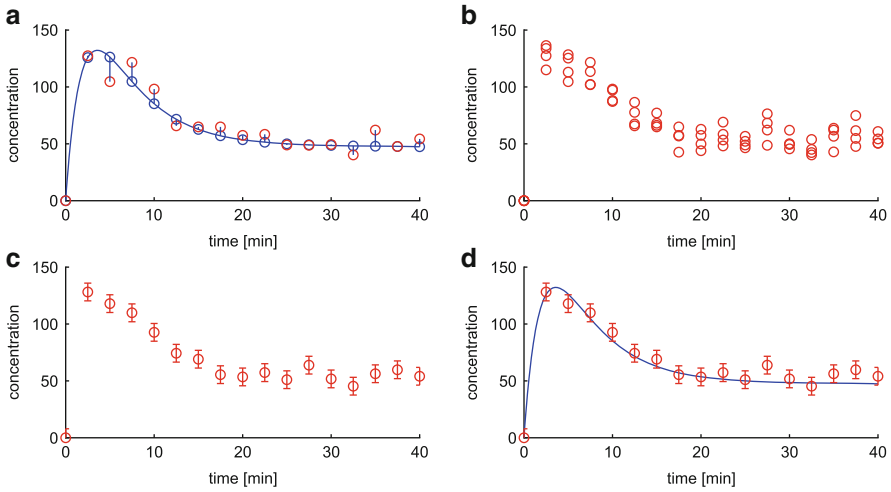

Fig. 3.3 Experimental observations of dynamic systems can often be decomposed in deterministic and stochastic components. (**a**) Time course data from a completely deterministic system, e.g., either a biological system with little variance or an artificial model (*blue line*) and the corresponding measurements with added stochastic effects (*red circles*). (**b**) To deal with stochastic effects, multiple measurements of each time point are performed, resulting in multiple values for each time point (*red circles*). (**c**) To quantify multiple replicates of experimental measurements, the mean value (*circle*) and standard deviation (*error bars*) for each time point are determined. (**d**) Comparison of the deterministic time course (*blue*) and the quantification of multiple observations (*red*)

If the experiment is repeated multiple times, x^{obs} instead contains the mean value of all observations. The deterministic time course data produced by simulating our ODE model is $x^{\text{sim}} = x^{\text{sim}}_{1\ldots n}$ and $x^{\text{stoch}} = x^{\text{stoch}}_{1\ldots n}$ is the difference between observation and simulation that we attribute to stochastic effects in our experimental setup:

$$\begin{aligned} x^{\text{obs}}_{1\ldots n} &= x^{\text{sim}}_{1\ldots n} + x^{\text{stoch}}_{i} \\ x^{\text{stoch}}_{i} &= x^{\text{obs}}_{1\ldots n} - x^{\text{sim}}_{1\ldots n} \end{aligned}$$

with $x^{\text{stoch}}_i \sim N(\mu, \sigma_i^2)$ and $\mu = 0$ we find (without proof) that the probability for every single observation is

$$P\left(x^{\text{stoch}}_i\right) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \mathrm{e}^{-\frac{\left(x^{\text{stoch}}_i\right)^2}{2\sigma_i^2}}$$

If we assume independence between the experimental variations at different time points (e.g., that a large variation at an early time point is not the cause of large variations at later time points), we can derive the probability for the entire time course by multiplying the probabilities of each single time point:

$$P\left(x^{\text{stoch}}\right) = \prod_{i=1}^{n} P\left(x^{\text{stoch}}_i\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} \mathrm{e}^{-\frac{\left(x^{\text{stoch}}_i\right)^2}{2\sigma_i^2}}$$

$P\left(x^{\text{stoch}}\right)$ is already a quantification of how well our model fits the experimental data; however, in practice this formula is rather inconvenient as it requires multiple evaluations of the exponential function and generally results in extremely small values that are too small to be represented in standard computational variables. However we can simplify the equation by dropping the normalizing factor $1/\sqrt{2\pi\sigma_i^2}$ and rescaling it on a logarithmic scale. Logarithmic rescaling allows us to remove the exponential function and replace the product over each time point with a sum. In addition, as we are generally interested in an equation that directly evaluates the quality of the simulation $x_{1\ldots n}^{\text{sim}}$, we substitute $x_i^{\text{stoch}} = x_{1\ldots n}^{\text{obs}} - x_{1\ldots n}^{\text{sim}}$ and derive the error function $\text{EF}\left(x^{\text{sim}}\right)$:

$$\text{EF}(x^{\text{sim}}) = \sum_{i=1}^{n} \left[\frac{\left(x_i^{\text{obs}} - x^{\text{sim}}\right)^2}{2\sigma_i^2} \right]$$

This formula is the sum of squared residuals, normalized by the variance of the observation. The value of the error function is zero if $x^{\text{obs}} = x^{\text{sim}}$ and positive otherwise; the closer the value of the error function is to zero, the better our simulated data fits to our experimental observations. If multiple replicates of the experiment have been performed, the experimental variance σ_i^2 can be calculated directly from the data points; otherwise it has to be estimated based on expert knowledge.

3.2.2 Parameter Optimization

Based on our introduction to ODE's in Sect. 3.2, it is obvious that the value of the simulated time course x^{sim} depends on a set of parameters. We will group all parameters of the ODE system into a vector of parameters we will call θ, where θ_i is the i-th parameter. For a mass action kinetic model, θ will usually consist of the kinetic rates of all reactions.

We will utilize the error function introduced in the previous section to find a value for θ that results in a good fit between simulation and experimental data. We do this by trying to minimize the value of $\text{EF}(x^{\text{sim}}(\theta))$. This process is called *parameter optimization*. As entire books have been written on the topic, we will limit us here to an overview over a few common techniques. All approaches introduced here are based on the same general idea; an initial value for θ is picked (either based on literature values or at random) and subsequently modified with the goal of improving the error function value. This is done iteratively until some kind of ending criteria is met. Common ending criteria include a fixed number of total iterations or a number of iterations without marked improvement in function value.

A very basic strategy is the *hill climbing* algorithm. In each step the local neighborhood of the current θ is explored. To do so, a candidate for a new

parameter set, $\hat{\theta}$ is generated by adding a value that depends on the exact implementation of the algorithm to a single element θ_i. If the change results in an improvement, θ is updated to the new value $\hat{\theta}$. Otherwise another candidate is generated by subtracting the same value from θ_i and again, the new candidate is accepted if it results in an improvement over the previous error value. This is then iteratively repeated for each element of θ until no further improvement is made. While this method is easy to implement, it has the disadvantage of frequently becoming stuck in local optima.

Another type of local methods are *steepest ascent methods*. In these methods, the next step in each iteration is chosen based on local evaluation or approximation of the first derivative of the model dynamics. This will usually result in a step that optimizes the improvement of the error function. While this sounds like a promising approach, it is limited by how well we can approximate the first derivate of the model. In addition, models that are determined by higher-order derivates will result in very small steps, causing long run times. As in hill climbing algorithms, there is a danger of getting stuck in local minima.

A common heuristic to deal with the issue of local optima are *simulated annealing* algorithms. The idea of these algorithms is that in each step a proposal is generated by randomly changing the current parameter vector. If the change results in a score improvement, it is always accepted. However, if the error value increases, the proposal is still accepted with a certain probability. This probability depends on a temperature value that start at high value and is then decreased according to a cooling schedule.

3.2.3 Choosing Significant Error Function Cutoffs

In the previous sections we discussed how to quantify the difference between the simulated time course data and the experimental observations, and gave an overview over parameter optimization techniques that can be used to minimize this quantified value in order to produce a good model fit. However, when we try to reduce models, we face the question whether a slightly worse error score justifies a significant simplification of the model. This leads to the question of *cutoff values* of the error function: Up to which error value can we say that a model reproduces our experimental data satisfactory?

One way to derive cutoffs for error values is based on the assumption that the observed data points are normally distributed around a time course generated by a deterministic dynamic. This is done using the χ^2 *distribution*. The χ^2 distribution calculates the probability that summing over a number k of squared, uniform normally distributed random variables results in a certain value. If we keep in mind that we assumed $(x_i^{\text{obs}} - x^{\text{sim}})$ to be normally distributed, this is exactly what we do in the error function $\text{EF}(x^{\text{sim}}) = \sum_{i=1}^{n} \left[\frac{\left(x_i^{\text{obs}} - x^{\text{sim}}\right)^2}{2\sigma_i^2} \right]$.

Based on the χ^2 distribution, we can estimate the *expected deviation* of an observation from the real value of the generating system and the resulting *expected error score per time point*. For example, we expect that 68% of all data points should be within one standard deviation of the deterministic time course (contributing an error value less than one error unit per time point) and 95% should be within two standard deviations (contributing an error value less than four error units per time point).

The exact calculation and interpretation of confidence intervals using this method is nontrivial and exceeds the scope of this chapter. However, as *a rule of thumb* based on these considerations, we expect a normalized sum of squared residuals lesser or *equal to the number of time points* to be almost always a rather good fit that explains most data points. Likewise, a score *larger than four times the number of time points* is almost always a bad fit that either completely misses some data points or shows a significant deviation from every single measurement.

It should be noted that, no matter how error cutoff values are derived, they should always be analyzed in the context of the experimental data and the biological system. It is very possible that a way to derive a cutoff value that works perfectly well for one set of data results in a cutoff that is to strict or to permissive in a different context. It generally makes sense to test multiple cutoff values and compare which value is closest to the interpretation of the experimental data in a biological context (Fig. 3.4).

3.3 Reduction of Graph-Based Models

In the context of this chapter, we focus on reducing a model by removing reactions that are not required to explain the observed dynamic behavior of the model. This reduces the degree of parameter under determination and helps to identify the core dynamics essential to the model.

In general, two different types of reactions compose a biochemical model. One is a set of *core reactions* that can be considered essential for a pathway. These reactions have either been confirmed in previous experiments, are established as a gold standard in literature, or are required in the model for structural reason, e.g., reactions that are important for model connectivity. We are generally not interested in reducing a model by removing core reactions.

In addition there are *auxiliary reactions* or *reduction candidates* R_{red}, reactions that are either of a hypothetical nature or of a detail level that might be inappropriate for the desired model. Examples for reduction candidates often include reactions that have been predicted based on binding domain analysis and PPI data, but have not been confirmed in vivo. For our analysis, any subset of R_{red} is a potential reduction of the model. The set of all subsets of a set X is also called the powerset of X, which we will denote $\mathrm{P}(X)$. Thus the entire space of possible reductions S_{red} is $P(R_{\text{red}})$. The size of a powerset grows exponentially to the

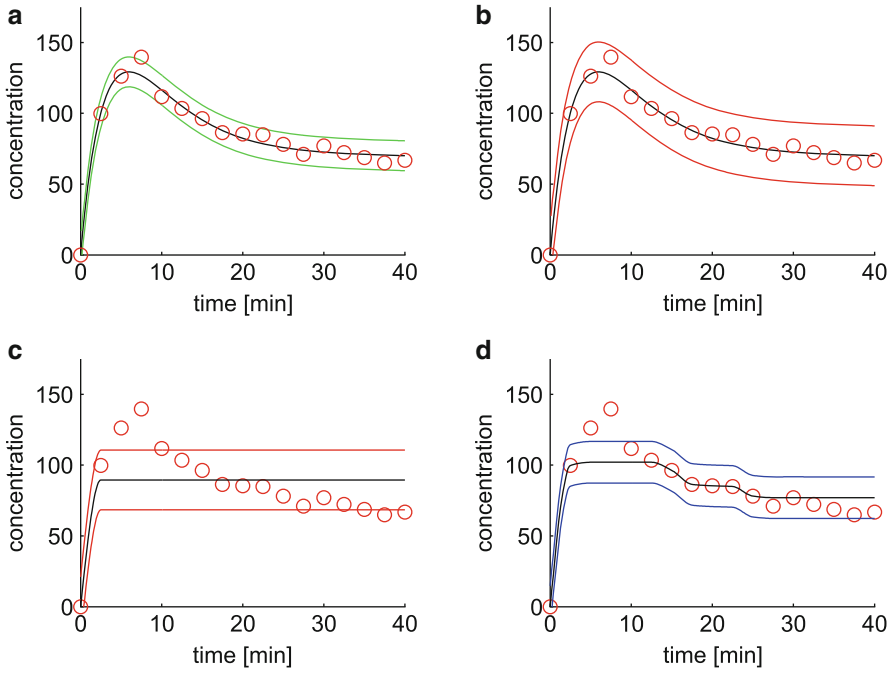

Fig. 3.4 Visualization of different kinds of cutoffs. (**a**) Cutoff based on one standard deviation (cutoff value equal to the number of time points). The *black line* is a possible simulation that satisfies the cutoff value, the green lines visualize the simulation +/− one standard deviation. Note that the borders visualized with respect to the simulated data; this is functionally identical to considering the borders with respect to the experimental data points. Note that it is not necessary that all points have to be inside visualized borders; if some data points are close to the *black line*, others might be outside the one standard deviation border. (**b**) Cutoff based on two standard deviations (cutoff value equal to four times the number of time points). The *red lines* visualize a two standard deviation border, (**c**) Cutoff based on two standard deviations, worst case fit. The simulated time course still results in an error value below the cutoff, despite two data points being outside *red borders*. In this case, the two standard deviation cutoff is too permissive, as an important qualitative feature of the data, the early activity peak is lost. (**d**) By analyzing the minimal error score at which the peaking behavior is lost, we derive a new error cutoff based on 1.4 standard deviations. The new worst case fit (*black line*) still shows a clear early peak

base 2. This means that the total number of possible reductions of a model is 2^n, where n is the number of reactions in R_{red}.

Most times we find that not all members of S_{red} are valid reductions, i.e., able to reproduce experimental data with a quality below a cutoff as explained in Sect. 3.2.3. The challenge lies in identifying which reduction candidates are not *supported by experimental evidence*.

It should be noted that parameter optimization for an ODE model is often a computationally expensive task that can take multiple minutes per attempt. It is therefore necessary to keep direct testing of reductions to a minimum. Instead,

we hope to verify/reject a large number of reductions *indirectly*. Since the complete reduction space grows exponentially with the number of auxiliary reductions, brute force checking of every reduction candidate will often take a prohibitively long time. *Heuristic strategies* need to be employed to speed up the identification reductions that are not supported by experimental evidence.

3.3.1 Indirect Model Verification and Rejection

Indirect model acceptance and rejection are based on a simple but powerful property resulting from the definition of our reduction framework. Based on the definitions given in Sect. 3.2, we find that removing a reaction from a model is identical to setting the associated kinetic parameter to zero. This implies that removing a reaction from an invalid model cannot transform the model into a valid model. Likewise adding reactions to a valid model cannot make this model invalid, as the newly added reactions could potentially have a kinetic rate of zero. This results in the following theorem:

Core theorem of set-based model reduction:

1. If a reduction is identified as valid, all reductions that are subsets of the valid reduction are also valid.
2. If a reduction is identified as invalid, all reductions that are supersets of the invalid reduction are also invalid.

These properties are essential to our design of reduction strategies. They allow us to accept valid and reject invalid reductions without the need to explicitly testing them. This is necessary, as the complete search space doubles with each additional reduction candidate, making explicit testing of all reductions impossible. We need to maximize the information gained indirectly in order to deal with the exponentially growing search space.

If we compare the indirect information gained from accepting/rejecting a reduction, we find that these will strongly differ between different candidates. Most of the time, reduction with few elements provide the most indirect information gain if they are rejected, as they are subsets of a larger number of reductions than large reductions. In contrast, large reductions provide the most information if they are accepted.

Direct testing of a reduction candidate is based on the analysis introduced in Sect. 3.2.2, 3.2.3, e.g., multiple parameter fitting attempts are started that attempt to find a model parameterization that explains the observed data with a quality below a cutoff. While we only introduce cutoffs that are motivated by statistical analysis of a given error model, cutoffs and error functions derived in a different way can also be utilized without changing the other aspects of model reduction. If a parameterization is found that results in an error value below a certain cutoff, the reduction is accepted as valid.

Unfortunately, we often find that the probability of accepting a large reduction picked at random is rather small; likewise, small reductions are more likely to be accepted than large reductions. This implies that, in order to analyze the reduction space efficiently, we need to focus on strategies that identify large reductions that are likely to be accepted or small reductions that are likely to be rejected with a higher than random frequency.

In addition to estimating the probability of rejecting/accepting a reduction, a second important aspect to optimize indirect information gain is keeping track of the reductions that have already been accepted and rejected. While accepting a large reduction will often result in a significant information gain, this is only true if only a small number of its subsets have already been accepted. In contrast, if most subsets of a large reduction candidate have already been accepted, the information gained by accepting the candidate is still small.

3.4 Topological Model Analysis

As already mentioned, the reduction space we wish to analyze contains all combinatorial subsets of the reduction candidates of a model. This space will frequently contain reductions that can be identified as unable to reproduce experimental data based solely on the topology of the reduced model (e.g., cases where biologically important states become disconnected from the rest of the model). Examples for such reductions are illustrated in Fig. 3.5. We call these reductions *topologically invalid* reductions.

In addition, it is possible that different sets of elementary reductions result in models that show identical dynamic behavior. For these *redundant* models, it is sufficient to test the validity of one reduction of the redundancy group and subsequently assign all other models the same validity. Examples for redundant models are given in Fig. 3.5.

To recognize topologically invalid and redundant models we use the concepts of *observability* and *controllability* and *activity*. While these properties are inspired by the concepts with the same names as utilized in systems engineering, it should be noted that we use significantly different versions. We illustrate these concepts for the ODE models, but they can be applied in a similar way to a large range of different models, including SDE models, Boolean models, and agent-based models.

3.4.1 Controllability

We use the property *controllability* to keep track of which inputs are able to control which intermediate- and output states. An input is said to control another state if changing the input results in a change of behavior for the second state. An example

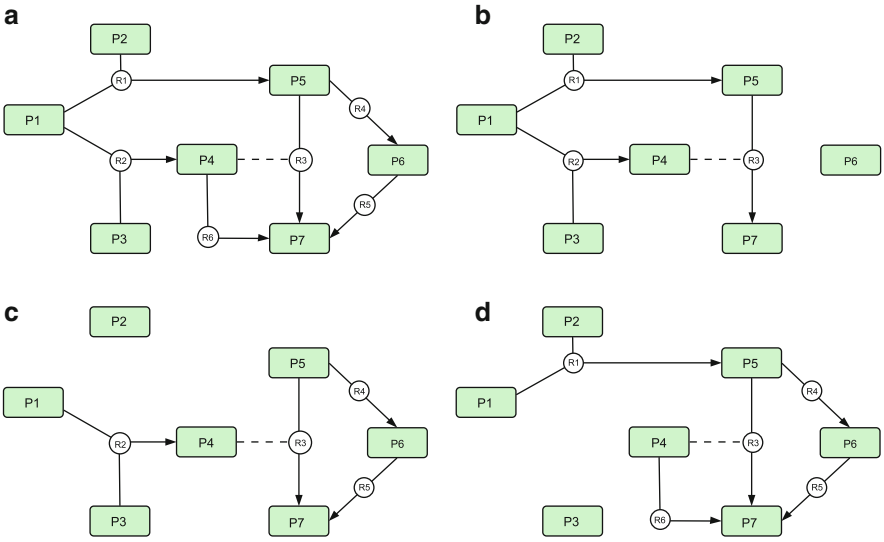

Fig. 3.5 Examples for valid, invalid, and redundant reductions. (**a**) is the original, unreduced model as introduced in Fig 3.1. (**b**) is a topologically valid reduction. Although both independent pathways that lead to the production of P7 have been removed, P7 is still produced by the cooperative pathway (reaction R3). If the cooperative reaction happens at a significantly higher rate than both indirect pathways, such a reduction can be realistic in a biological context. If this is the case, the model behavior is determined primarily by the cooperative activation mechanism. (**c**) is an invalid reaction. The direct activation of P4 has been removed. In addition, the dimerization of P1–P2 has been removed, so that P5 is no longer produced. This results in a situation where P7 can no longer be produced, rendering the entire pathway inactive. Therefore, model (**c**) is an invalid reduction. (**d**) This reduction is still able to produce P7 through the independent activation of P5, but can no longer produce state P4. Thus it is a valid reduction. However, it still includes two reactions that depend on P4 (R3 and R5). These reactions will never have a value larger zero, rendering them obsolete. A reduction of model (**d**) that would also remove R3 and R5 would act identically to model (**d**) without obsolete reactions. Therefore, we call model (**d**) redundant

for this is that during a gene knockout experiment, all other genes that are up- or down regulated are controlled by the gene knocked out. Models that do not connect the genes we observe to be experimentally controlled by the gene knocked out can be rejected without time-consuming parameter optimization attempts. For the type of ODE models that we consider, the control of states is *propagated* between the states by the reactions, as states do not directly depend upon each other. Reactions in turn are completely determined by a set of states that is specific for each reaction.

Intuitively, we find that an Input State SI:

- Controls itself by definition.
- Controls a reaction R_n if $(\text{Subs}_R(R_n) \cup \text{Cat}_R(R_n))$ contains any controlled states, i.e., if any substrate or catalyst of is R_n controlled.

– Controls a state S_n if $(\text{Subs}_S(S) \cup \text{Prod}(S_n))$ contains any controlled reactions, i.e., if any reaction that either consumes or produces S_n is controlled.

These intuitive definitions are lacking in so far as they allow for a state's controllability to depend recursively upon itself. While such a situation is easily recognized by a human researcher, a computer-based analysis needs to explicitly account for this possibility. It should be noted that these conditions are only *necessary*, but not *sufficient* to confirm control. As we only consider topological criteria, parameterizations can exist for which we do not observe control relations despite the topological conditions being fulfilled. This is the case if, e.g., the kinetic parameter of a reaction that is required for a control relation is set to zero.

3.4.2 Observability

The property *observability* is closely related to controllability. We call a state observable if changes to the state (either at a certain time point or to the initial conditions) can be recognized in the states that we are able to measure experimentally. Similar to the limitations imposed in the analysis of controllability, we are again limited to necessary conditions and have to avoid recursive dependency. We will use observability to identify reactions that are unimportant for the model dynamics we observe and can be removed. If any reactions are identified that are not observable, the model is by definition *redundant*, as a model that removes these would behave identically with respect to our experimental observations.

A state S_n:

– Is observable by definition if S_n is an output state
– Is observable if $(\text{Subs}_S(S_n) \cup \text{Cat}_S(S_n))$ contains any observable reaction

A reaction R_n:

– Is observable if $(\text{Subs}_R(R_n) \cup \text{Prod}(R_n))$ contains any observable state

3.4.3 Activity

We find that simulations of biological processes frequently contain only a few states that start with an initial concentration greater zero, whereas the majority of states will have an initial condition of zero. This can result in situations where multiple states and reactions will always have a concentration of zero, for all possible experimental setup. We use the property of *activity* to determine if a reduced model contains any reactions that will always have a value of zero. If this is the case, the model is *redundant*. In addition, if any output state is not active, the model is *invalid* (Figs. 3.6 and 3.7).

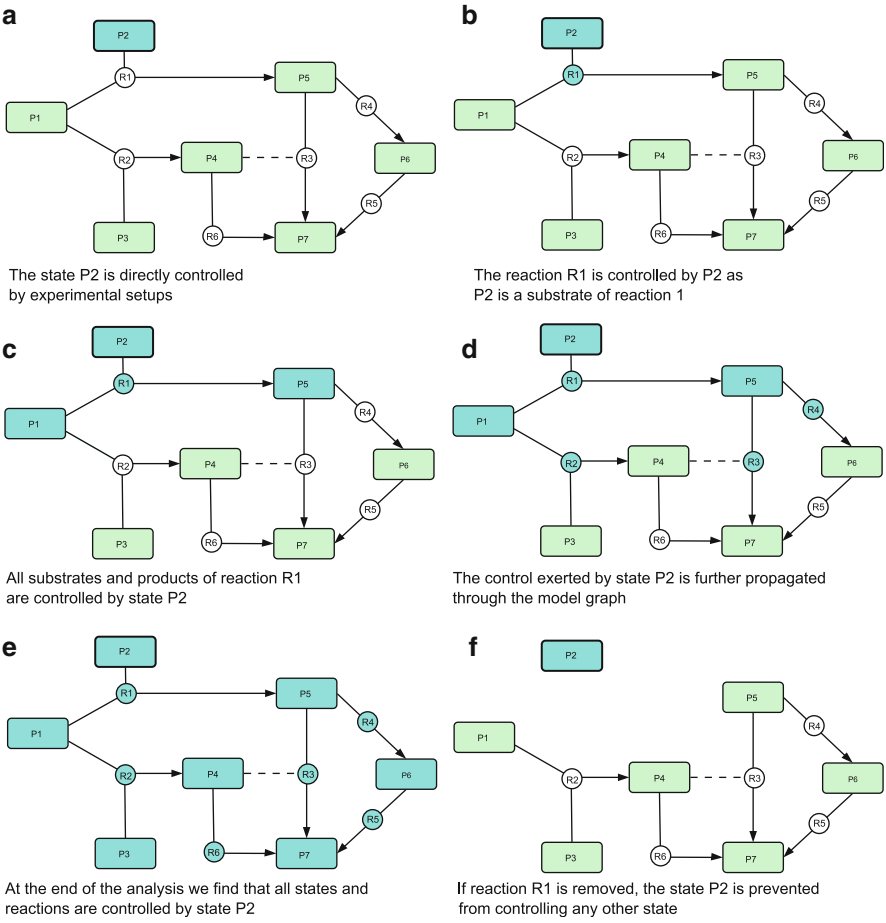

Fig. 3.6 Illustrating the concept of controllability. (**a**) State P2 can be controlled directly by choosing different experimental setups. (**b**) The reaction R1 is controlled by P2, as P2 is a substrate of R1. (**c**) Control is propagated from R1 both *forward* (to its products) and *backward* (to its substrates). (**d**) Applying this propagation iteratively allows us to analyze the remaining model. (**e**) We find that state P2 exerts control over the entire model. (**f**) If the reaction R1 is removed, P2 loses its entire ability to control the model

A state S_n is active:

– If it has an initial concentration larger than zero
– If $\mathrm{Prod}(S_n)$ contains any active reactions

A reaction R_n is active:

– If all states in $(\mathrm{Subs}_S(S_n) \cup \mathrm{Cat}_S(S_n))$ are active

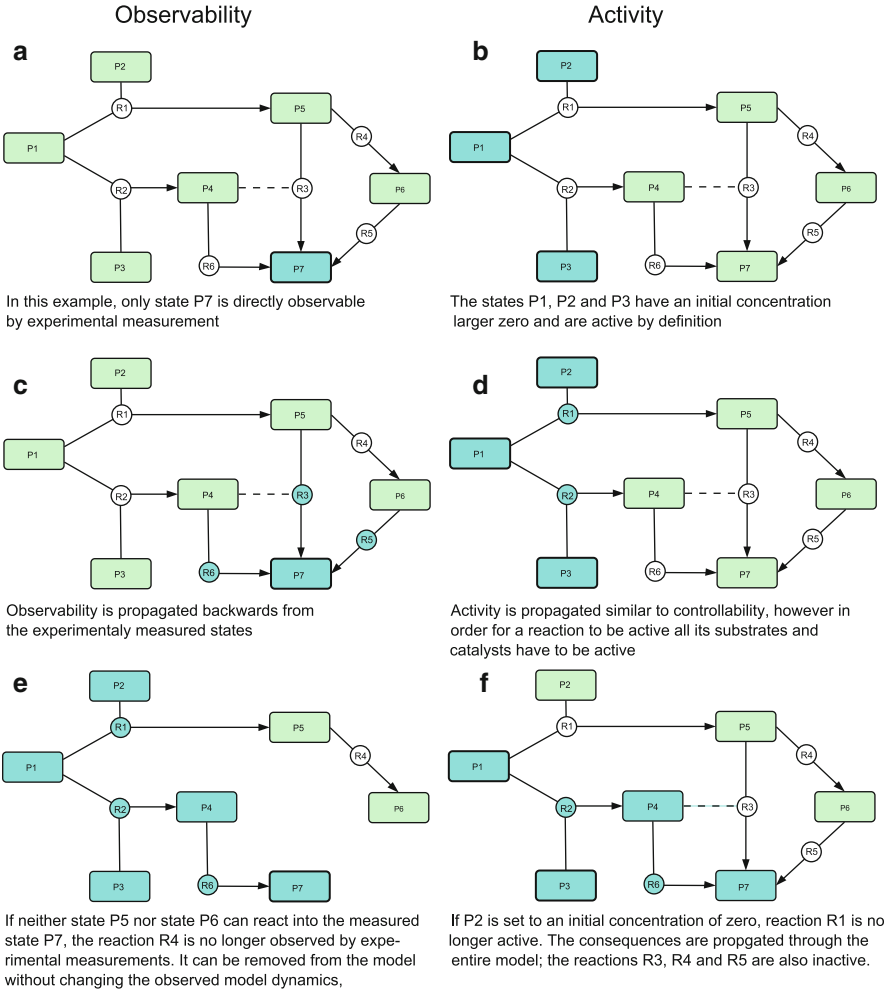

Fig. 3.7 Activity and observability. Activity (*right side*) and observability (*left side*) are analyzed similar to activity; however, some important differences exist. (Observability) In contrast to Controllability, the initially observed states are the model outputs. A reaction is observable if either one of its products or substrates is observable. It will propagate this observability to all its substrates and catalysts (but, in contrast to Controllability, not to its products). (Activity) All states with an initial concentration larger than zero are initially active. Frequently, the set of initially active states will either be identically to or a superset of the controlled states. However, it is insufficient that one substrate or catalyst of a reaction is active to propagate activity to a reaction. Instead, all substrates and catalysts have to be active. Reactions in turn only propagate activity in a forward fashion to all their products, but not to their substrates or catalysts

Summary

Controllability
Input State SI:

Controls itself by definition.
Controls a reaction R_n if $(\text{Subs}_R(R_n) \cup \text{Cat}_R(R_n))$ contains any controlled states, i.e., if any substrate or catalyst of is R_n controlled.
Controls a state S_n if $(\text{Subs}_S(S) \cup \text{Prod}(S_n))$ contains any controlled reactions, i.e., if any reaction that either consumes or produces S_n is controlled.

Observability
A state S_n:

Is observable by definition if S_n is an output state
Is observable if $(\text{Subs}_S(S_n) \cup \text{Cat}_S(S_n))$ contains any observable reaction

A reaction R_n:
Is observable if $(\text{Subs}_R(R_n) \cup \text{Prod}(R_n))$ contains any observable state

Activity
A state S_n:
Is active if it has an initial concentration larger than zero
Is active if $\text{Prod}(S_n)$ contains any active reactions

A reaction R_n:
Is active if all states in $(\text{Subs}_R(R_n) \cup \text{Cat}_R(R_n))$ are active

Modeling Implications
A model that contains any inactive or unobservable reactions is *redundant*.
A model that contains any inactive output state is *invalid*.
A model that violates any experimentally established input/output dependencies is *invalid*.

3.5 The Reduction Graph Data Structure

In order to better visualize the concepts introduced in Sect. 3.3, it can be helpful to further analyze the structure of the reduction space in a way that does not depend on the model we want to reduce. We already characterized the reduction space as the powerset of all reduction candidates. We can utilize this by analyzing the inclusion structure, i.e., the subset/superset relation between its elements. This inclusion structure can be visualized by a special graph called Hasse diagram. Hasse

diagrams are utilized to visualize Partially ordered sets, of which powersets are one example.

Any reduction candidate R_1 that is a subset of another candidate R_2 will be considered an ancestor of R_2. If, in addition, R_1 has exactly one element less than R_2, R_1 will be called the direct ancestor or parent. Inversely, R_2 will be called a descendant of R_1 if R_2 is a superset of R_1 and a direct ancestor or child of R_1 if R_2 has exactly one element more than R_1.

To generate a Hasse diagram, we construct a graph in which each reduction candidate (including the empty set) is assigned one node. Every node is connected to its direct descendants by a directed edge. The result is a hierarchical, directed acyclic graph starting from the empty reduction (which has no incoming edges) to the complete reduction (which has no outgoing edges). All elements of one hierarchical level remove exactly the same number of reduction candidates.

The theorem in Sect. 3.3.1, can be reinterpreted in this context:

1. If a reduction is identified as valid, all nodes that have a directed edge leading to this reduction can also be marked as valid reductions
2. If a reduction is identified as invalid, nodes that can be reached from this reduction can also be marked as invalid

Based on this interpretation, we can reinterpret the analysis of the reduction set as a path-finding problem in a graph: we are interested in all nodes that can be reached by directed paths that start at the empty reduction set node. This allows us to utilize *path-finding algorithms* designed for different graph-based problems with little adaptions.

When analyzing the reduction graph of a model, it becomes obvious that most valid and invalid reductions can be verified/rejected indirectly. Based on the considerations in the previous section we find that every valid reduction that also has a valid descendant can be validated indirectly. Likewise, any invalid reduction that has an invalid ancestor can be rejected without the need for explicit model checking.

Only two types of reductions have to be tested explicitly. Invalid reductions that have only valid parents have to be rejected by direct testing. We will call these reductions *minimal invalid reductions*. The description "minimal" is used to clarify that all models that remove only a subset of a minimal invalid reduction are valid. Similarly, valid reductions that have no valid children will be called *maximal valid reductions*. To completely analyze the reduction space of a model, it is both necessary and sufficient to find both the set of *minimal invalid reductions* and the set of *maximal valid reductions*. However, in reality, we find that direct identification of these sets is rarely possible. Instead, our goal is to find heuristic search strategies that minimize the direct testing performed for reductions that are neither maximal valid nor minimal invalid (Fig. 3.8).

3.6 Search Strategies

The strategies we will introduce in this chapter all utilize a candidate list that is updated with each accepted or rejected reduction step. If a reduction is accepted, all its direct descendants that have not already been rejected are added to the end of the candidate list. If a reduction is rejected all its descendants that are currently in the candidate list are removed. The order in which candidates are picked from the list (e.g., oldest first, newest first...) determines the exact algorithm. These strategies are strongly similar to path-searching algorithms for graphs. A common property of all candidate-generating strategies is that every reduction directly tested (except the empty reduction set) will always have at least one valid parent.

3.7 Basic Search Strategies

A *Breadth first search* strategy is one of the basic approaches to analyzing the reduction graph. It is implemented as a candidate generating strategy that always picks the first element of the priority queue, i.e., the oldest element as new candidates are added to the end of the queue. This is also called a first in first out priority strategy.

There are strong similarities between this search strategy and the a priori algorithm for frequent item set mining as introduced by Agrawal et al. (1994). Several results regarding the best- and worst-case runtime of the a priori algorithm can be transferred to the Breadth first search strategy. Like the a priori algorithm, a breadth first search operates in a semideterministic fashion. If multiple breadth first runs are started, they will always explicitly test the same reductions, potentially in a varying order.

Based on the analysis of the a priori algorithm, we can also make observations regarding the number of unnecessary explicit tests performed. A breadth first search will only test nodes whose ancestors have all been verified, i.e., accepting a node during a breadth first search will never result in additional information gain. Vice versa, all nodes that will be rejected will be minimal invalid reductions, as they have no invalid ancestors. The result is that the breadth first search will always explicitly test all valid nodes, but in turn only test those invalid nodes that cannot be avoided to be tested.

Based on this we find two general applications; if a model either has only a very small number of valid reductions or rejecting an invalid reduction is on average significantly more costly than accepting a valid reduction.

A *depth first* search strategy can be implemented very similar to a breadth first search, with the difference that the last item of the candidate queue is picked at each step, i.e., the item that has been added most recently (a last in first out priority strategy). However, the resulting search dynamics will strongly diverge from the behavior of a breadth first algorithm. In general, the performance of a depth first search will vary significantly between different restarts.

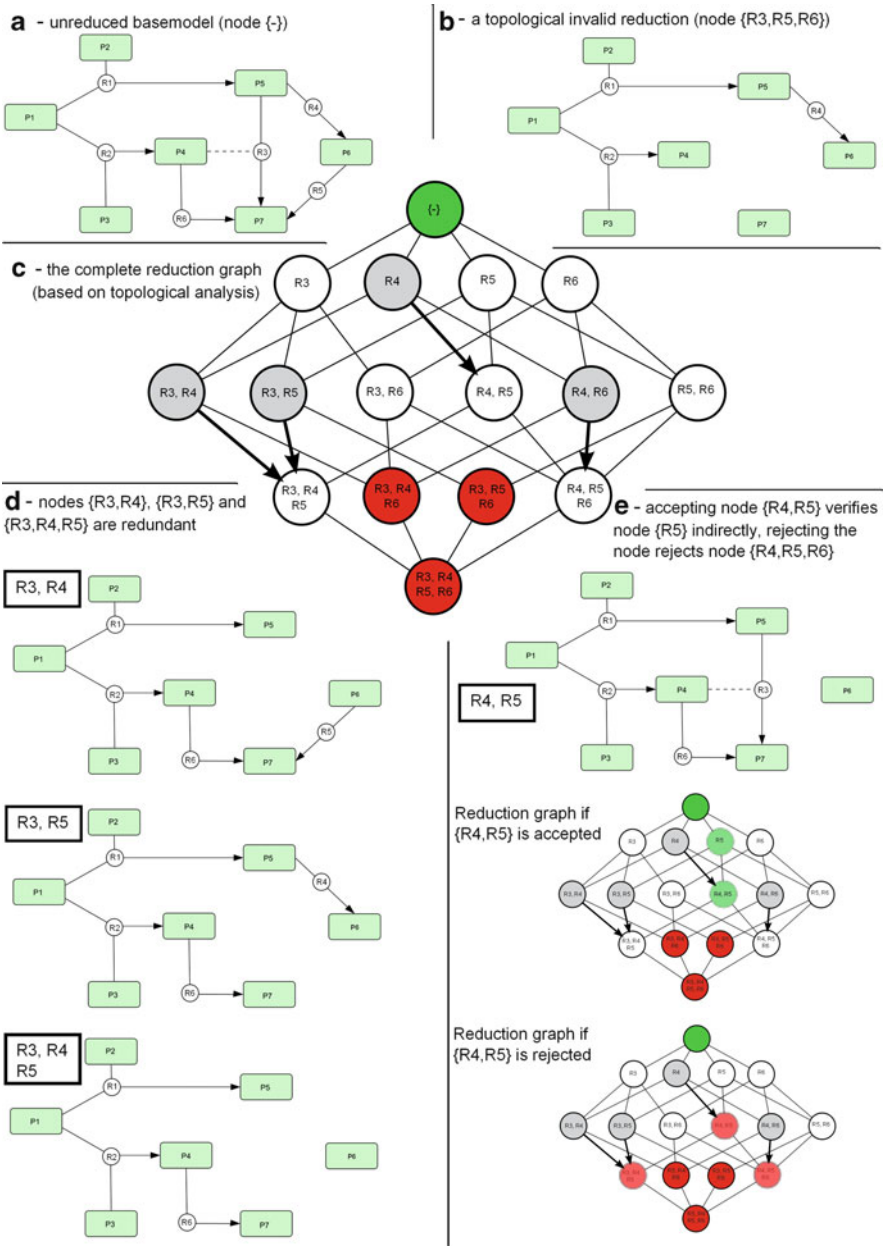

Fig. 3.8 The reduction graph data structure. (**a**) The unreduced base model is mapped to the root node (−) or empty reduction of the reduction graph. It is always valid. (**b**) The reduction B is topologically invalid. It is mapped to node (R3, R5, R6). (**c**) The initial reduction graph for the base model (**a**), after topological analysis but prior to starting a reduction run. Note that only the reactions R3–R6 are reduction candidates, R1 + R2 are considered established reactions. *Red nodes* are topologically invalid, *gray nodes* are redundant. Redundant nodes are connected

A depth first search will most of the time attempt to verify reductions larger than the last accepted reduction. Only if all descendants of the current reduction are rejected will the depth first search start to trace back to earlier reductions. Ideally, the search will find large valid reductions early in its course, indirectly verifying a large number of valid reductions, thus significantly outperforming the Breadth first search. However, if a small invalid reduction is missed, a depth first search can end up getting "stuck" rejecting all its descendants in an unfavorable order. The following example illustrates this problem and compares the approach of the depth first to the breadth first approach.

Both breadth- and depth first approaches will perform very badly if the space of valid reductions is structured in certain ways. The number of verifications required by breadth first approaches grows proportionally to the size of the solution space even if the solution space is structured very regular. In contrast, depth first searches can get stuck in irregular-shaped solution spaces. Therefore it makes sense to include a *random walk*-based strategy as a benchmarking baseline. In such a strategy a random member of the priority queue is chosen at each step. Interestingly, we find that such a random walk-based search will often outperform both breadth- and depth first approaches. This illustrates that the problematic cases discussed for breadth- and depth first approaches occur with significant frequency in modeling applications, and that strategies to deal with these cases are required (Fig. 3.9).

3.8 Hybrid Switching Approach

If we compare the performance of breadth-first, depth-first, and random walk search during an reduction run as illustrated in Fig. 3.10, we find recurring properties. At the start of the reduction run, the depth-first strategy will often outperform both alternative strategies. The depth first strategy initially identifies large valid reductions with a higher frequency then both alternatives. This offers a significant indirect information gain by indirectly verifying a large number of smaller reduction candidates. However, it subsequently gets stuck rejecting a large number of reductions with very small indirect information gained for each rejection.

In contrast, the breadth first search exclusively gains indirect information by rejecting invalid reductions, as has already been discussed. This will often result in

Fig. 3.8 (continued) by an *arrow* to the reduction that is obtained by removing all obsolete elements from them. If a search encounters such a redundant node, it automatically skips to the node indicated by the *arrow* as it is a unique representative of this redundancy group. (**d**) Both (R3, R4) and (R3, R5) are redundant. The unique representative of this redundancy group is the reduction (R3, R4, R5). (**e**) Explicitly testing the reduction (R4, R5) can have two results. If it is accepted, this will indirectly verify (R5). If it is rejected, this will indirectly reject (R3, R4, R5) and (R4, R5, R6). The resulting reduction graph for either case is illustrated

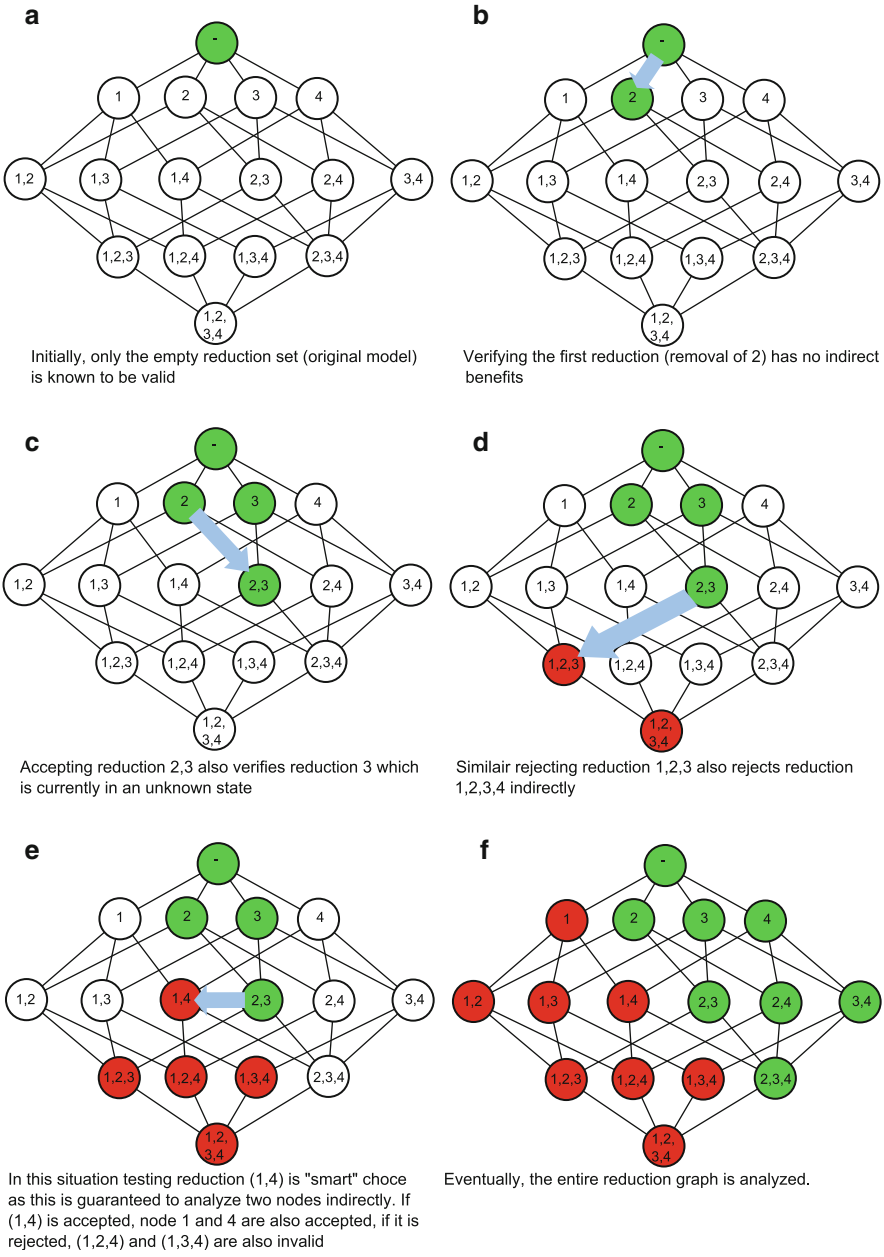

Fig. 3.9 Direct and indirect validation and rejection in the reduction graph. (**a**) Most search strategies will start at the root node of the reduction graph. (**b**) Verifying that (2) is a valid reduction does not provide indirect information gain. (**c**) Verifying node (2,3) also verifies node (3). Note that node (2) has already been verified, so it does not count as indirect information gain, despite being an ancestor of node (2,3). (**d**) Similar to (**c**), rejecting (1,2,3) results in indirect

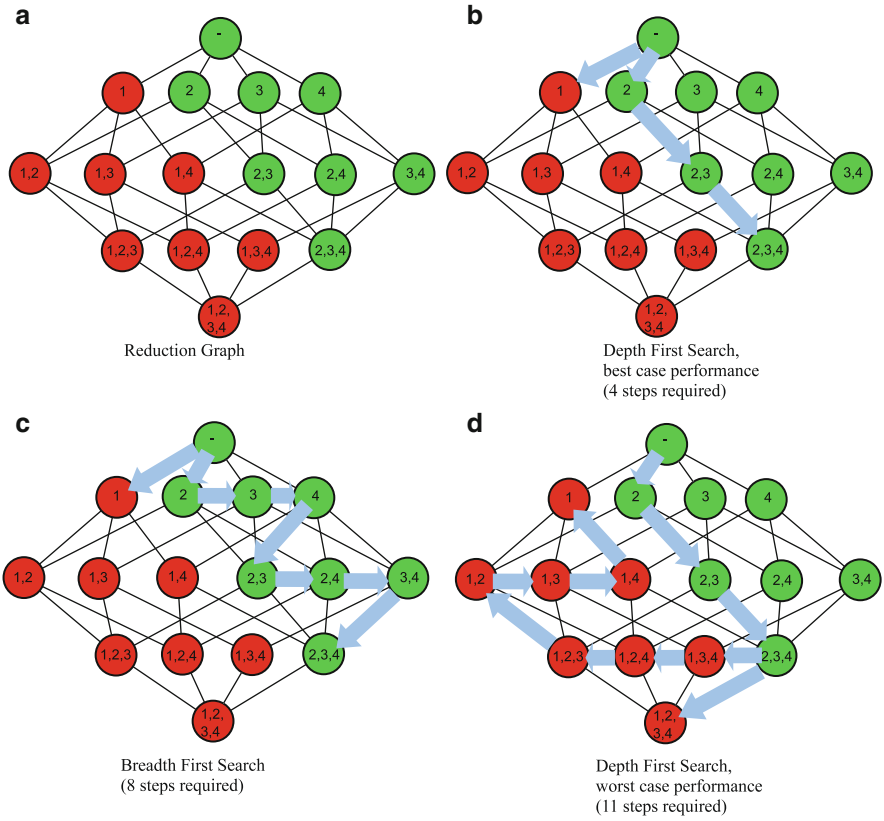

Fig. 3.10 Comparing depth first and breadth first random search. (**a**) Completely analyzed reduction graph. (**b**) Best case of a search trace for a depth first search. The search initially rejects (1) and indirectly rejects all its descendants. It then proceeds to analyze the remaining, valid part of the reduction graph in only four steps. (**c**) A breadth first search will always identify (1) as an invalid reduction during its first few steps. However, it will need to verify all remaining valid reductions explicitly, resulting in a performance that is worse than the best case of a depth first search as illustrated in (**b**). (**d**) However, if a depth-first analysis misses the small invalid reductions and directly wanders to node (2,3,4), the rejections of the invalid reductions will be done by backtracking from a large valid reduction. This case results in the worst-case performance that is worse than the breadth first search

Fig. 3.9 (continued) information gain by indirectly rejecting (1,2,3,4) (**e**) In the situation as displayed in **d**, the analysis of node (1,4) a next step is a smart choice. Either accepting or rejecting it will result in the indirect analysis of two nodes. Note that no parent of (1,4) has been analyzed; therefore, only a candidate picking, but not a candidate generating search strategy would be able to utilize this reasoning to skip to node (1,4). In this example, node (1,4) is rejected, resulting in the additional rejection of (1,2,4) and (1,3,4). (**f**) Once the complete graph is analyzed, we know all valid reductions of the initial model

a slow initial phase, where a large number of small valid reductions are explicitly tested. In exchange, the problem of long stretches of subsequent rejection with little information gain is completely avoided, resulting overall in a stable performance that has smaller information gains then the alternatives during early phases and larger information gain towards the end.

From comparing all three basic strategies, we already know that on average, the good early performance of the depth-first strategy and the good late performance of the breadth-first strategy are insufficient to set off the respective disadvantages when problematic situations are encountered. We also find that the advantages and disadvantages of depth- and breadth-first search supplement each other.

An alternative to the weighted random walk we call hybrid switching approach is to initially start with a DFS that is interrupted as soon as a certain number of rejections has been reached. Once this has happened, either a new DFS is started (that would use a path different from the initial DFS) or the DFS phase is stopped and a BFS is started to analyze the remaining reduction space. Criteria that are possible to decide the time of switching from DFS to BFS include the information gained during the last DFS run or the number of remaining unknown reductions.

3.9 Application Example: Reducing a Model of the CD95 Pathway

Regulation of cell death decisions via CD95 signaling involves complex dynamics of the involved pro- and anti-apoptotic proteins, e.g., procaspase-8 and c-FLIP, and their cleavage products. These sometimes interact in surprising, non-intuitive ways. A signal that induces cell proliferation and survival at low concentrations can induce apoptosis at higher concentrations, thus resulting in opposing effects depending on whether a threshold is met (Lavrik et al. 2007). To understand not only the qualitative level of these regulatory mechanisms, but the details of the molecular interactions resulting in such a threshold behavior, researchers have begun developing quantitative signaling models (Fricker et al. 2010). While these models are currently able to illustrate the molecular dynamics encountered, they typically suffer from indeterminacies stemming from either over specified models or biologically relevant alternative architectures. This reduces their value in model-based prediction, as parameter uncertainties will often directly result in uncertain and ambiguous predictions. In this section, we will illustrate how model reduction can be utilized to improve our understanding of the processes happening during this signaling and to derive new models that better represent these processes.

To analyze the role of c-FLIP cleavage in apoptosis induction, we have used a model describing the apoptotic branch of the CD95 signaling pathway as implemented in (Fricker et al. 2010). This model is illustrated in Fig. 3.11. While the model explains the interaction of c-FLIP in an intuitive way, it is considerably underdetermined. Many intermediate states of the pathway can only be measured as

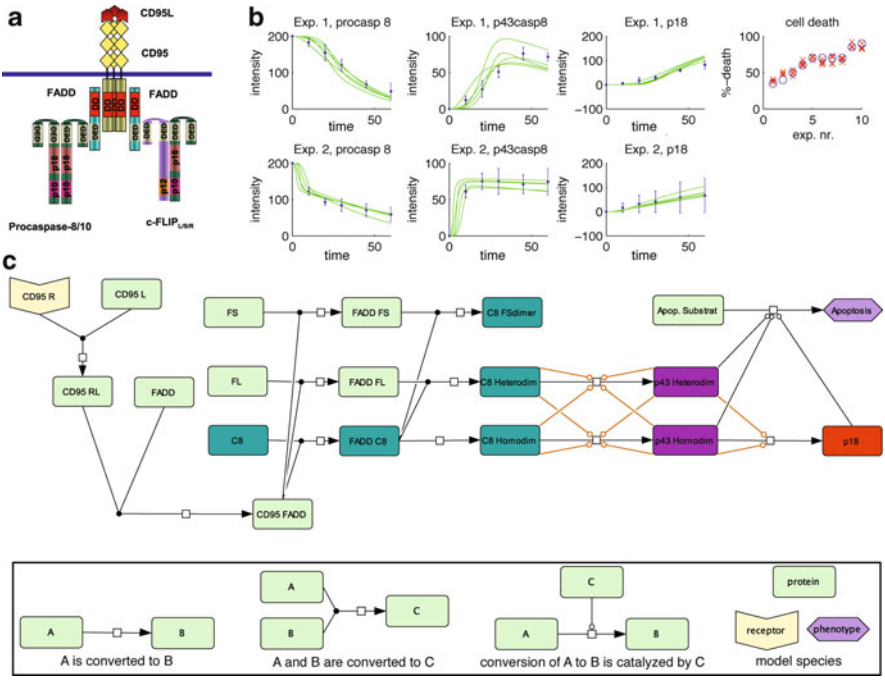

Fig. 3.11 Biological background, experimental data, and computational model of the CD95 pathway. (**a**) The DISC (Death Inducing Signaling Complex) is formed at the membrane. Its most important parts are the CD95 ligand/receptor complex and FADD (Fas Associated Death Domain) protein. Both Procaspase-8/10 and c-FLIP$_{S/L}$ are recruited by and bound to this complex and further processed and activated. (**b**) Experimental measurements of various molecules in the CD95 pathway during stimulation experiments. For protein measurements, the *x*-axis denotes time while the *y*-axis denotes relative intensity. *Blue error bars* are the measured points, *green lines* are simulations by the pathway model (**c**) with different parameter sets. All simulations have similar overall quality. For the cell death measurements, the *x*-axis denotes the number of the experiments, each *blue circle* is a separate experiment. The *y*-axis denotes the number of cells that had died by the end of the experiment. Red *x*'s are again the result of simulating the CD95 model with different parameter sets. (**c**) The CD95 model. Equally colored species were measured as one experimental concentration; dissociation reactions were omitted for clarity. Reactions that are candidates for removal have been highlighted in red. Our goal is to find model reductions that are roughly as good as the simulations illustrated in (**b**), but contain fewer of the reactions marked as reduction candidates

groups and a significant number of potential reactions cannot be observed directly as they occur in membrane-localized complexes that are difficult to measure in an experimental context.

The model is roughly divided into three parts. The first part of the model simulates the binding of the extracellular CD95L (CD95 ligand) to CD95R (CD95 receptor) (Suda et al. 1993). The activated receptor recruits FADD. Bound FADD multimerizes and thus creates the membrane-localized DISC (Death Inducing Signaling Complex), denoted as CD95 FADD in the model (Kischkel et al. 1995).

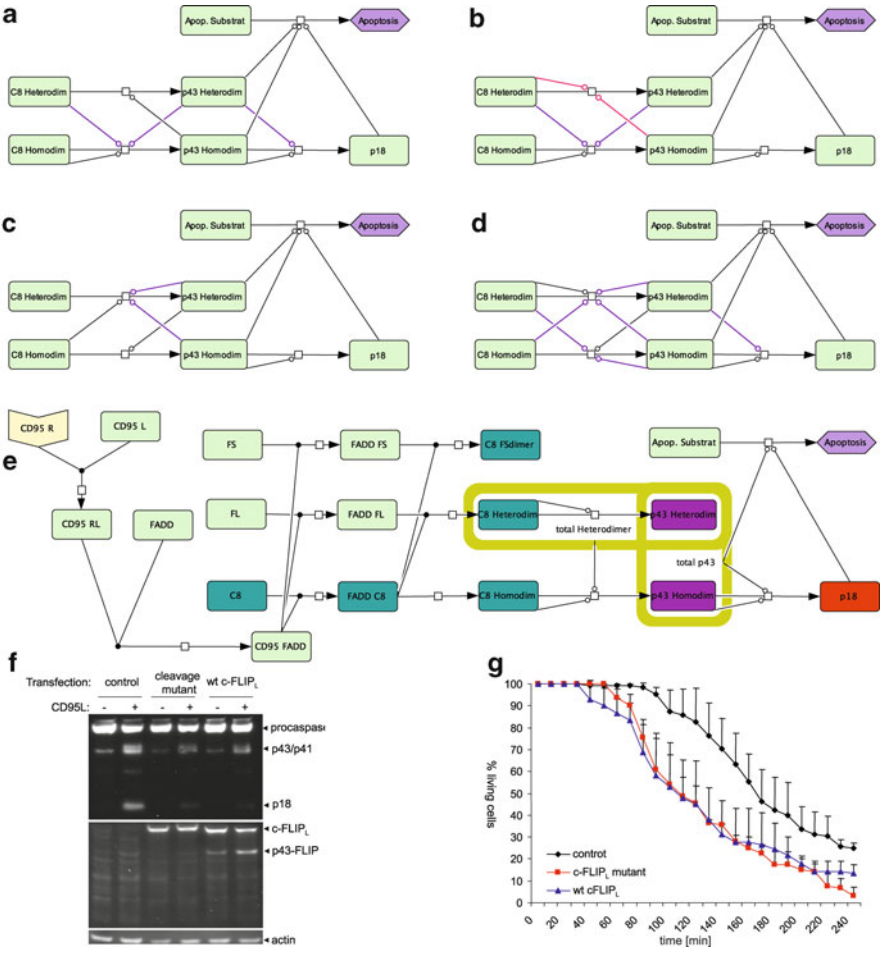

Fig. 3.12 Model reductions of the full CD95 signaling model and CD95 consensus model. The 14 reduced models are summarized in (**a**–**d**) by including any one (**c**, **d**) or two (**a**) edges of the same color or one edge of each color (**b**). A recurring pattern is that all reductions include the activation of the C8 homodimer by either the C8 or the p43 heterodimer. This illustrates that clearly the interaction of homo- and heterodimers is an essential part of the pathway dynamics and that limiting the model to homo–homo and hetero–hetero interactions is not a valid approach. The minimal reductions summarized in (**c**) can be considered questionable in a biological context; it seems unlikely that the C8 homodimer can activate the C8 heterodimer but not itself while the C8 heterodimer cannot act as catalyst at all. This illustrates that our current set of experimental data is insufficient to completely characterize the CD95 pathway in a satisfying fashion. The same holds true for the reductions summarized in (**d**); it seems biologically questionable that C8 homodimers show no catalytic activity. Based on these considerations we derive a new consensus model based on the minimal reductions summarized in (**a**) + (**b**). Both (**a**) and (**b**) retain the autocatalytic activation of the C8 homodimer. In addition, both models have to include the activation of the C8 homodimer by either the C8 or the p43 heterodimer. As either reaction is fine, it can be reasoned that current data suggests that both reactions are dynamically very similar. The resulting hypothesis that the C8 homodimer is activated by both the C8 heterodimer and the p43 heterodimer with

The second part of the model summarizes the binding reactions of procaspase-8 and c-FLIP isoforms to the DISC which results in the formation of different types of dimers. Three types of catalytically active dimers are formed: procaspase-8-homodimers (called C8 dimers), procaspase-8/c-$FLIP_L$ heterodimers (called C8 heterodimers), and procaspase-8/c-$FLIP_S$ heterodimers (Neumann et al. 2010).

The third part of the model focuses on the activation of the different procaspase-8 dimers. C8 dimers and C8 heterodimers are proteolytically activated by (spatially) neighboring dimers and further processed. In contrast the procaspase-8/c-$FLIP_S$ heterodimer is not processed further. The procaspase-8 part of the dimers is cleaved into the active form p43/p41, resulting in p43/p41 homo- and heterodimers. p43 homodimers are further processed into the caspase-8 tetramer containing the cleavage product p18.

The proteolytically activated heterodimers p43/p41 and p18 forms of procaspase-8 can contribute to the progression of apoptosis by activating various downstream effector caspases. This effect is summarized as the cleavage of apoptosis substrate. The model state "apoptosis" is measured by rate of cell death experimentally.

The part of the model that is most severely underdetermined is the activation of both C8 homodimer and C8 heterodimer. It is known that this reaction has to be catalyzed; however, C8 homodimer, C8 heterodimer, p43 homodimer, and p43 heterodimer are all candidates as possible catalysts for both reactions. The initial model therefore includes four activating reactions for each C8 homodimer and C8 heterodimer, one for each possible catalyst. A similar situation occurs for the activation of apoptosis substrate; here, both p43 heterodimer and p43 homodimer are potential candidates for catalyzing this reaction.

We applied the model reduction approach as introduced in the previous chapter to the unreduced base model of the CD95 pathway, using all reactions that activated either the C8 homodimer, the C8 heterodimer, or the apoptosis substrate as reduction candidates. The total set of reduction candidates contained 11 reactions, resulting in $2^{11} = 2{,}048$ possible reduction sets. Redundancy and validity analysis reduced the unknown model space by about 45% to a total of 1,158 nonredundant reductions.

Running the reduction search identified 237 reductions as valid, the 921 other reductions were invalidated. The model space can be characterized by 14 minimal valid reductions and eight maximal invalid reductions, as illustrated in Fig. 3.12. All minimal models reproduce the experimental data with an error score comparable to the unreduced base model. All valid reductions included the activation of

Fig. 3.12 (continued) the same rate is consistent with modeling results. Additional experiments with a c-$FLIP_L$ cleavage mutant were performed and compared to the behavior of wildtype c-$FLIP_L$ (**f**). In the mutant cell line the activation of C8 heterodimer to p43 heterodimer is blocked, no p43 heterodimer is produced. Quantification of cell survival (**g**) shows that cell survival rates are the same for wildtype and cleavage mutant cells, supporting the hypothesis that the C8 heterodimer activation does not change its catalytic influence on C8 homodimer activation.

apoptosis substrate by the p43 heterodimer. The largest valid reduction removes six reactions, any reduction removing seven reactions is invalid.

Every valid reduction includes the activation of the C8 homodimer by either the C8 heterodimer or the p43 heterodimer. In turn the autocatalytic activation of the C8 homodimer or the activation of the C8 homodimer by the p43 heterodimer is also included. The minimal satisfactory solution for these two conditions, the activation of C8 homodimer by the p43 heterodimer, but not by either itself or the C8 homodimer is part of various valid reductions. Indeed, the autocatalytic activation of the C8 homodimer is retained only in 4 out of 14 minimal reductions, although there is a strong biochemical evidence reported showing formation of C8 homodimers at the DISC.

The interpretation of these properties confirms that the role of c-FLIP$_L$ cannot simply be reduced to that of an inhibitor of CD95 signaling. Instead, the C8 heterodimer acts as a catalyst for the activation of C8 homodimers, either directly or in the activated p43 heterodimer version. However, various reported mechanisms of caspase-8 activation cannot be verified based on the experimental data currently available. This mainly concerns the catalytic activity of C8 both in homo- and heterodimer form. We expect that the autocatalytic activation of C8 homodimers is an essential process in the CD95 apoptosis signaling; yet our experimental data does not reflect this.

References

Agrawal R, and Srikant R Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487–499, Santiago, Chile, September 1994.

Fricker N, Beaudouin J, Richter P, Eils R, Krammer PH et al (2010) Model-based dision of CD95 signaling dynamics reveals both a pro- and antiapoptotic role of c-FLIPL. J Cell Biol 190 (3):377–389, Intro to CD95 Modeling

Kischkel FC, Hellbardt S, Behrmann I, Germer M, Pawlita M et al (1995) Cytotoxicity-dependent APO-1 (Fas/CD95)-associated proteins form a death-inducing signaling complex (DISC) with the receptor. EMBO J 14(22):5579–5588, Forming of DISC

Lavrik IN, Golks A, Riess D, Bentele M, Eils R et al (2007) Analysis of CD95 threshold signaling: triggering of CD95 (FAS/APO-1) at low concentrations primarily results in survival signaling. J Biol Chem 282(18):13664–13671, Intro to threshold

Neumann L, Pforr C, Beaudouin J, Pappa A, Fricker N et al (2010) Dynamics within the CD95 death-inducing signaling complex decide life and death of cells. Mol Syst Biol 6:352, Role of cFLIPl

Suda T, Takahashi T, Golstein P, Nagata S (1993) Molecular cloning and expression of the Fas ligand, a novel member of the tumor necrosis factor family. Cell 75:1169–1178, CD95R/L Binding

Systems Biology Toolbox for MATLAB: A computational platform for research in Systems Biology, Bioinformatics, 22(4):514–515, 2006.