



Freie Universität Berlin

Fachbereich Mathematik und Informatik
Diskrete Biomathematik

Logical modeling of uncertainty in signaling pathways of cancer systems

Kirsten Thobe

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)

Berlin, März 2017

Erstgutachterin: Prof. Dr. Heike Siebert
Fachbereich Mathematik und Informatik
Freie Universität Berlin

Zweitgutachterin: Prof. Dr. Christine Sers
Institut für Pathologie
Charité Universitätsmedizin Berlin

Tag der Disputation: 19. Mai 2017

Danksagung

Ich möchte mich ganz herzlich bei meiner Betreuerin Heike Siebert bedanken, die mich über die letzten Jahre motiviert und geleitet hat, und damit diese Arbeit ermöglicht hat. Die nicht müde wurde, mir Grundlagen und Zusammenhänge zu erklären und oft mehr Vertrauen in mein Urteilsvermögen hatte als ich selber. Auch möchte ich meiner Zweitbetreuerin Christine Sers danken, die mit ihrer Forschung zusammen mit Christina Kuznia Antrieb und Inspiration für meine Arbeit waren und mich, mit meinen zwei linken Händen, sogar in ihr Labor einluden.

Auch möchte ich mich bei meinen Kollegen der Arbeitsgruppen Siebert und Bockmayr bedanken, für die vielen Gespräche zu Mittag, das Korrekturlesen von Skripten und leckerem Gebäck bei Seminaren. Insbesondere danke ich meinem PostDoc Hannes Klarter, der mir geduldig diskrete Mathematik erklärte und mit seiner Software Grundlage für diese Arbeit war. Ebenso möchte ich Adam Streck für unsere produktive Zusammenarbeit danken und der vielen Hilfe mit seiner Software. Außerdem geht ein ganz großer Dank Katinka Becker und Firdevs Topcu-Alici, die mir über die Jahre mit ihrem Lachen und offenen Ohr zu Freundinnen geworden sind.

Der IMPRS-CBSC möchte ich für die viele Unterstützung danken, sei es finanziell um zu interessanten Konferenzen zu fahren, an hilfreichen Kursen und interessanten Retreats teilzunehmen, aber auch für die Orientierung und Hilfe vor allem von Kirsten Kelleher. Darüber hinaus habe ich in der IMPRS Familie viele tolle Menschen und Freunde kennengelernt, die mich in den letzten Jahren unterstützt haben.

Schließlich möchte ich meinen Freunden, sei es die Osaftler in der Heimat oder meine Freunde in Berlin, und meiner Familie, insbesondere meinen Eltern und Schwestern, danken: für das Teilen von Frust und Freude, aber vor allem für den bedingungslosen Rückhalt.

Contents

1	Introduction	1
2	Theoretical Background	7
2.1	Topology of the model	7
2.2	Regulatory mechanisms	8
2.3	Dynamical behavior of a model	9
2.3.1	Update strategy	9
2.3.2	Characteristics of dynamics	11
2.4	Model pool	12
2.5	Data processing	13
2.5.1	Discretization	14
2.5.2	Temporal logic	15
2.6	Model checking	17
2.7	Software	19
2.7.1	TomClass	20
2.7.2	Tremppi	23
3	Toolbox for evaluating uncertainty in biological systems	29
3.1	System initialization	29
3.1.1	Model boundaries & resolution of entities	30
3.1.2	Activity levels	33
3.1.3	Interactions and labels	34
3.1.4	Regulation constraints	36
3.2	Objective formalization and system adaptation	37
3.2.1	Investigating crosstalk between models	37
3.2.2	Finding driver mutations	38
3.2.3	Testing the effect of drugs	40
3.3	Data formalization	42
3.3.1	Incorporating genotype information	42
3.3.2	Steady state assumption	43
3.3.3	Choice of strictness	44
3.3.4	Monotonicity of data	45
3.3.5	Qualitative observations	45
3.3.6	Transfer properties to a higher dimension	46
3.4	Model pool analysis	46
3.4.1	Statistical analysis	47
3.4.2	Exact analysis	47

4	Implementation in Tremppi	49
4.1	System initiation & objective formalization as PKN	50
4.2	Data formalization for filtering the generic model pool	51
4.3	Analysis of the specific model pool	54
5	Investigating cell line specific EGFR signaling	59
5.1	Signaling in cancerous cells	59
5.1.1	MAPK pathway	60
5.1.2	PI3K pathway	61
5.2	EGFR signaling pathway study	63
5.2.1	Motivation	63
5.2.2	Model building and data formalization	65
5.2.3	Filtering and analyzing the cell line specific pools	69
5.2.4	Discussion	74
6	Crosstalk analysis between MAPK and PI3K signaling	77
6.0.1	Biological background	77
6.1	Crosstalk analysis using literature data	79
6.1.1	Model building and integration	79
6.1.2	Crosstalk analysis	81
6.1.3	Discussion	86
6.2	Signaling in RCC cells: role of Sorafenib and crosstalk	87
6.2.1	Model definition and objective formalization	88
6.2.2	Data processing & formalization	90
6.2.3	Analysis of cell line specific model pools	92
6.2.4	Discussion	98
7	Unraveling the regulation of mTORC2	101
7.1	Biological background	101
7.1.1	Conflicting studies on mTORC2 regulation	102
7.2	Results	104
7.2.1	Model building from literature	104
7.2.2	Data formalization	106
7.2.3	Model pool analysis	111
7.2.4	Experimental design	113
7.3	Discussion	116
8	Conclusion	121
	Bibliography	127
	Appendix A Supplementary data and figures	139
	Appendix B Abstract	149
	Appendix C Ehrenwörtliche Erklärung	151

Introduction

“ *All models are wrong, but some are useful.* ”

— **George Box**

Research Professor of Statistics

Models are idealized, simplified representations of phenomena and are used in a wide range of disciplines, from architecture and engineering to mathematics and biology, where e.g. a physical model is a scaled design of a building, mathematical models are built to optimize flows through machines, and a mouse as a model organism that is genetically modified to simulate human diseases and their treatment. In general, constructing a model requires the simplification of the system s.t. main characteristics are preserved. Then, model building itself as well as an analysis of the model can lead to new useful insights about the original system.

In systems biology, mathematical modeling of biological processes was shown to be valuable to increase their understanding [105]. With the availability of high-throughput genomic and proteomic data, the focus of research is shifting from grasping the function of individual proteins to unraveling how the many proteins interact together in a complex web of signaling, regulatory, structural and metabolic pathways in the cell [46].

Overview of different modeling approaches In order to understand and predict the behavior of a cell, we require a system level of understanding of the wiring of the pathways of the cell [51]. For this aim, modeling methods from different disciplines, such as engineering and computer sciences, are applied to biological systems to represent different facets. Figure 1.1 visualizes a selection of modeling approaches in systems biology ordered by their level of detail.

At the lowest level of detail, basic information about the wiring between components and their connections is gathered from correlations by statistical data-mining. More detail can be added by applying Bayesian statistics, which account for conditional dependencies between components. Thus, connections can have different weights indicating the influence of the components on each other [46]. These statistical

approaches are commonly used for top-down modeling, which means that the model is directly inferred from the data set with the aim of finding the simplest model that is accurate. An issue of these approaches is that they tend to ignore prior knowledge from the literature, and thus causal information of the system. An advantage is that the statistical approaches can account for uncertainties in the system and are able to deal with large systems.

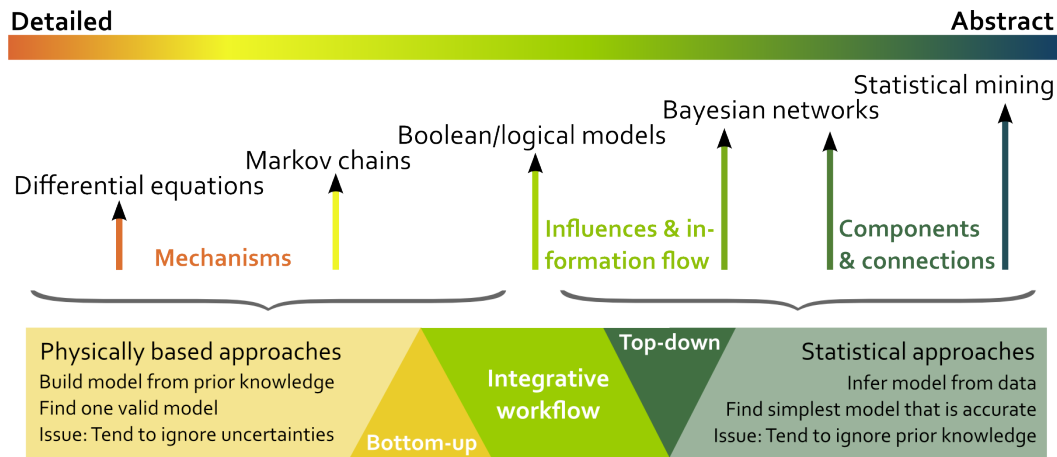


Fig. 1.1. Selected modeling approaches in systems biology. Different modeling approaches are listed from detailed, physically based approaches to abstract, statistical approaches. The detailed approaches usually build models in bottom-up fashion whereas abstract approaches mostly derive a model from data in a top-down manner. Figure adapted from [46]

At a high level of detail, systems of differential equations, most commonly ordinary differential equations (ODE), model dynamical processes such as diffusion or biochemical reactions, simulating the location or concentration of each component in the system over time. For this aim, very detailed information about kinetic constants, binding coefficients or transport rates as parameters are required. Less detailed information is necessary for Markov chain models where the production, loss and conversion of molecular species are probabilistic processes [46]. These detailed approaches are physically based and built in a bottom-up manner. This means that the models are built from prior knowledge about molecular processes such as binding or transport in the cell. The aim is to find one model that fits observed data and thereby is validated. These modeling approaches hardly are able to include uncertainty towards the topology and the regulatory mechanisms into their analysis, since they have to deal with unknown parameter values. Thus, all mechanical aspects are predefined and only the uncertainty in parameters is usually estimated by sensitivity analysis [66]. However, for uncertainties in the wiring of the system, a comprehensive analysis would be too exhaustive at this level of detail.

The medium level of abstraction is represented by the Boolean/logical formalism, which models the wiring of the system as a set of components with qualitative

on/off levels, called states, and the molecular interactions between them as connections [48]. Moreover, the functional relationships between components are defined as logical rules without requiring quantitative parameters. The simulation of the qualitative dynamics of the system can then link input combinations to output decisions such as cell-fate, which was shown to deliver valuable results for signaling processes [105, 57, 35]. Despite the fact that logical models are mostly built in bottom-up manner, there are also Probabilistic Boolean networks that infer the models from data [83]. Here, I want to use an integrative approach that is based on prior knowledge from literature, but accounts for uncertainties in the wiring and the functional relationships between the components (see Fig. 1.1).

Modeling cellular signaling In this thesis, I am especially interested in exploring signaling pathways, which form a complex molecular machinery in the cell to sense inputs and react with the appropriate output (see Fig. 1.2). For this purpose, the cell has a variety of sensors called receptors that specifically recognize stimuli and pass the signal to a protein network, which functions as an information processor [59]. The outcome influences the cellular machinery by producing new proteins through gene expression, secreting new stimuli, causing changes in the cytoskeleton or even triggering cell fates like growth or death.

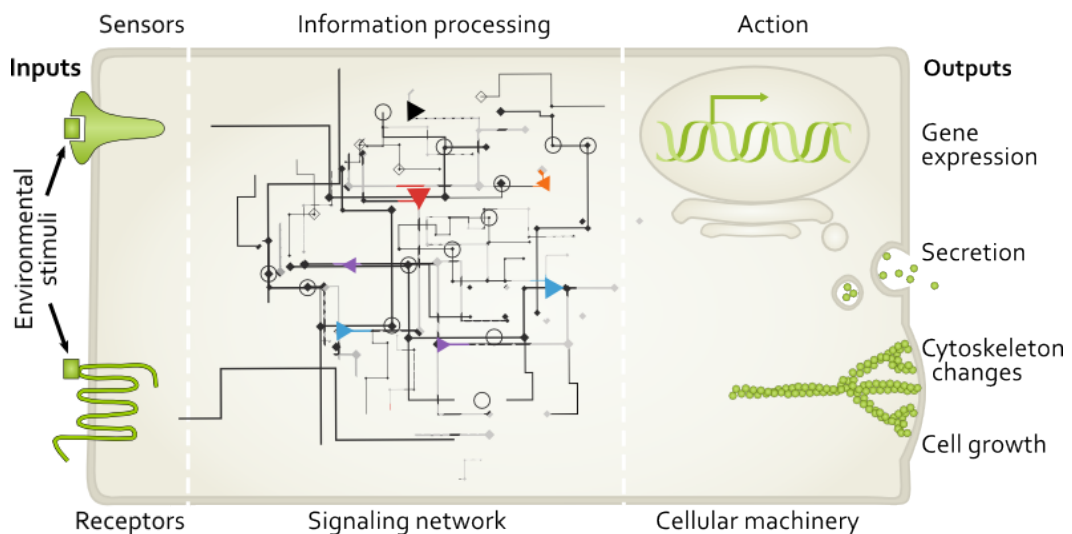


Fig. 1.2. Scheme of signal propagation in cells. Environmental stimuli activate receptors on the cell surface to trigger a reaction of the cellular machinery. The signaling network processes the different input combination to decide for the appropriate output (adapted from [59], network designed by Harryarts-Freepik.com)

For modeling signaling networks, three principles are important: modularity, robustness, and use of recurring circuit elements [3]. Modularity means that each pathway as a unit is designed for a certain purpose. Robustness can be realized by

having parallel mechanisms that control a certain function in order to compensate for malfunctions. This is realized through interconnections between the pathways, called crosstalk. Finally, recurring circuit elements are motifs that compose a unit, which encode functionalities such as feedbacks.

Traditionally, signaling networks are investigated based on the principle of modularity, which means that individual pathways connect a receptor through a chain reaction with a specific set of target genes. Thus, these individual pathways are relatively well described. In contrast, the principle of robustness through crosstalk is less well understood [1], and therefore often uncertain. However, cancer cells often use these crosstalk connections to escape treatment, therefore including for robustness is of interest [77].

Logical modeling of uncertain systems The wiring of cellular signaling in human cells is still far from understood. Especially in cancer cells this problem is enhanced by mutations [74]. When modeling an uncertain system, one option is to build a model based on assumptions. However, another option is to build every possible model that arise from the uncertainty and compare their performance. Depending on the modeling formalism, building every possible model can become computationally challenging, e.g. finding parameters for one ODE model is already a hard problem usually also rife with uncertainty. Our group developed a logical modeling workflow [98] to create and analyze many possible topologies and mechanisms of biological systems (Fig. 1.3) using efficient software [53, 90].

In our approach, we use a bottom-up model building to include all available information about the system and categorize them as certain or uncertain information. Here, uncertainty means that there are controversial results in the literature on the topology and/or on the regulatory mechanisms. Then, all possible topologies and mechanisms are enumerated to a generic model pool. In the second step, we compare the dynamics of the models in the pool with new data and thereby reduce the number of models in a top-down fashion to determine specific subpools (see Fig. 1.3). With this approach, we can analyze tens of thousands of models efficiently.

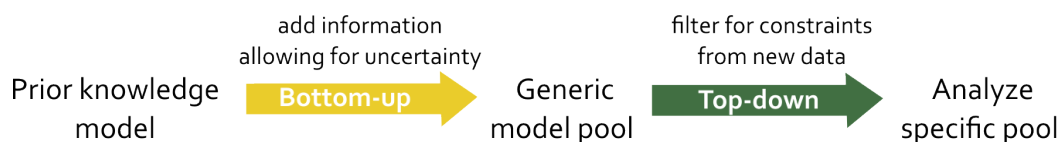


Fig. 1.3. Workflow for the modeling approach used here. First a generic model pool is created from all available information including uncertainty. Then, the pool is filtered for data to find specific subpools, which can be analyzed for new information.

Other logical modeling approaches that incorporate uncertainty are available, e.g. CellNetOpt or an Answer Set Programming based approach by Videla et al. being similar to our approach [96, 102]. These tools differ in a number of aspects from our approach, in particular, they train models according to an optimality criteria rather than considering the full set of consistent models. Also, these tools focus on steady state responses, which poses a problem in signaling systems. These systems often contain negative feedbacks that hamper the measurement of steady states and cause non stable behavior in the model dynamics. For this reason, CellNetOpt splits the model dynamics into an early and late steady state, where the first one excludes the feedback and the second one includes it. However, this separation requires detailed information about the time-scales of the modeled system, which we do not presuppose. Other related work was done by Martin et al., where Boolean dynamics of genetic regulatory networks were inferred from data, but this is a purely top-down approach without including prior knowledge [67].

Aims and structure of the thesis In this thesis, I expanded the workflow (Fig. 1.3) to a toolbox, focusing on the interface between biology and mathematics. Here, the formalization of biological information and objectives as well as the biological interpretation of the outcome of the method are the focus of this work. In Chapter 2, I first describe the theoretical background for logical modeling, introduce the concept of model pools and the method model checking for comparing these pools to data. I used two different model checking tools in this thesis, which are described briefly in the last section of Chapter 2.

The main methodical work is presented in Chapter 3, where I describe how the general workflow is expanded to a comprehensive toolbox for evaluating uncertainty in biological systems. Since the workflow has only been applied to toy systems before, my aim was to explore which kind of biological systems are interesting to analyze, what kind of questions can be addressed and how these questions can be formalized? Since the focus is the application to cellular signaling processes in cancer cells, three interesting objectives were identified that can be analyzed with the workflow: investigating crosstalk between models, finding mutations, and testing the effect of drugs. In order to analyze these systems for data in the next step, experimental measurements need to be formalized, which includes discretization of quantitative data, interpretation of temporal information such as steady state assumptions and consideration of qualitative information such as mutations. The last section in Chapter 3 then deals with the question of, given a statistical and exact analysis tool, how can the results of a model pool analysis be interpreted to extract

biological information? Then, Chapter 4 gives a short introduction, on how to model this toolbox in one of the model checking software Tremppe.

In the following Chapters 5 to 7, the toolbox is applied to four different case studies (three of them published [92, 98, 97]). Here, the biological system I investigated were two prominent cancer signaling pathways, but each case study investigates a different aspect of the system. In Chapter 5, I tested and compared the approach to a published study, where the signaling network of 6 cancer cell lines was investigated in a rich data set. Here, the cell line specific wiring of the network was explored for different temporal constraints to determine the impact of these constraints on the results.

Chapter 6 investigates the crosstalk between two well-known signaling pathways systematically. According to the methodology described in the toolbox, I aimed to introduce crosstalk while preserving validated behaviors of the single pathways. The first study in the chapter is based on literature information and models the healthy system. The results then form the basis for the second study for two renal cancer cell lines. Here, the crosstalk as well as the effect of a cancer drug on the signaling system is explored using data from our collaboration partners and analyzed with focus on possible drug targets and crosstalk combinations.

In the last chapter, the workflow is applied to a small signaling network, where the wiring of one specific component is uncertain with conflicting studies in the literature. Here, we systemically collected all proposed hypotheses and supporting data from the literature to unravel the conflicting information. Chapter 7 also shows how our approach can be used for experimental design to propose further studies that could clarify the control of that component.

Theoretical Background

In Chapter 2, the theoretical background on logical modeling is given. The generation of a model pool is described formally, which is required for the modeling process suggested in the workflow in Figure 1.3. The formal description is based on the definitions in the dissertation of Hannes Klarner [53]. Moreover, in order to compare experimental observations with discrete simulations of the models, a formalization of data as temporal logics is presented. Finally, model checking software to efficiently apply these formulas is introduced.

2.1 Topology of the model

In logical modeling, the topology of a biological system is represented as a directed graph $\mathcal{R} = (V, E, l)$, called *interaction graph* (IG). This graph contains nodes $V = \{1, \dots, n\}$, which represent the *components* of the system. These components are biologically functional entities from single genes or proteins, to complete cells or organs, depending on the desired level of abstraction. The activity of this functionality is encoded in discrete values $\mathbb{N}_0 = \{0, 1, 2, \dots\}$, called *activity levels*, depending on how many different functionalities of the component are measurable and relevant. In case, only one functionality is assigned to all components of the system and a *Boolean network* (BN) with $\mathbb{B} = \{0, 1\}$ is given, where 0 means the functionality inactive and 1 stands for active. The further definitions are given for BNs.

By assigning activity levels to every component of the network, the *state* of the system s is defined by $s : V \rightarrow \{0, 1\}, \forall v \in V : s(v) \in \mathbb{B}$. Here, the notation of a state is specified as a sequence in the order of V . For example, if the model has four components $V = \{v_1, v_2, v_3, v_4\}$ with Boolean activity levels, a possible state would be $s = 1001$, meaning that $s(v_1) = 1, s(v_2) = 0, s(v_3) = 0,$ and $s(v_4) = 1$. The combination of every activity level of each component gives rise to the *state space* $\mathcal{S} = \prod_{v \in V} \mathbb{B}$.

In the interaction graph, an edge $e \in E \subseteq V \times V$ is called *interaction* and represents a regulation of one component by another. The nature of this regulation is described

by the edge label $l : E \rightarrow \{+, -\}$. If, e.g. a protein A is physically binding another protein B and thereby activating it (Fig. 2.1), then A is called *regulator* of B and the influence is translated into an activating edge from A to B with the label $+$. An activation means that an active state of A causes at some point an increase in the activity level of B. The second possible influence is inhibition, where an active state in the regulator causes at some point a decrease in the activity level of the target component. In a network that is not Boolean, each interaction is active for certain ranges of values $\theta(u, v) \subseteq [1..Max(u)]$ where $Max(u) \in \mathbb{N}_0$ is the maximum activity level of u .

2.2 Regulatory mechanisms

Though the interaction graph provides information about the wiring of a network, it is not sufficient to describe the mechanical processes behind the interactions. Often a component is affected by more than one regulator, e.g. having two activating influences. These influences might act independently or are both necessary to activate the target, thus for representing the biological mechanism it is crucial to express these dependencies. There are two common ways of describing the regulatory mechanisms of the components, either by giving the logical formula or parametrization function.

The conditions under which a component is active can be expressed as conjunction and disjunction of the regulators. For a full description of the network, a formula f for every component $v \in V$ is required to define every state of the system. These formulas are constructed as expression as follows:

$$f ::= 0 \mid 1 \mid u \mid \neg f \mid f_1 \vee f_2 \mid f_1 \wedge f_2$$

where $u \in V$ describes a component, $\neg f$ the negation of f , $f_1 \vee f_2$ the disjunction and $f_1 \wedge f_2$ the conjunction of the expressions f, f_1, f_2 . For sparse networks, these formulas are intuitive to understand and already give information about the underlying mechanisms, e.g. for the toy model in Fig. 2.1.

An alternative way to define the regulatory mechanism is to enumerate every possible regulatory context of a component and to assign the corresponding effects in terms of activity values. This assignment is called *parametrization* and is equivalent to generating a truth table for each component as can be seen for the toy example in

Figure 2.1 c. Here, the formal definition of the regulatory mechanism for each $v \in V$ by its regulators $v^- = \{u \in V \mid (u, v) \in E\}$ is the *partial parametrization*

$$K_v : 2^{v^-} \rightarrow \mathbb{B} = \{0, 1\}.$$

Here, an element $U \in 2^{v^-}$ corresponds to the row in the truth table in which each $u \in U$ is active and all others are inactive. The value assigned to that row is $K_v(u)$. U is sometimes called a regulatory context of v . A special case are so-called *input nodes* where $v \in V$ is defined as a node that only has itself as a predecessor $v^- = \{v\}$ with an activating self-regulation.

Having K_v specified for every component, the full parametrization of the system $K = (K_v)_{v \in V}$ is given.

2.3 Dynamical behavior of a model

With a parametrization K or the logical formulas F , the dynamical behavior of the network \mathcal{R} can be described. For this aim, the transition of the model from one state to another as a simulation generates its behavior over discrete time steps. These state transitions are described by $\rightarrow \subseteq S \times S$ applying the logical formulas or parametrization functions as an update of the system's state.

2.3.1 Update strategy

There exist different update strategies, the main ones are synchronous, asynchronous, and sequential update. Sequential update creates transitions according to a pre-defined sequence of components, which usually is based on information about the timing or order of events from the biological system. Therefore, it requires specific prior knowledge about the dynamical behavior. The synchronous update assumes that all components in the system change to their next state, before any component can evolve further in time. Therefore it cannot account for fast and slow processes in one model. However, it is biologically unrealistic that all components work completely synchronized, especially since the components can have different level of abstraction. Still this update strategy is the most commonly used, because the dynamics are relatively simple and computationally manageable even for larger networks. The reason for simplicity is that both sequential and synchronous update produce deterministic dynamics, that is, each state has only one possible transition to the next state.

An alternative update strategy was proposed by Thomas et al., where they employ asynchronous update of the state [99]. Here, only one component can change its value at a time. For a state $s = (s_1, \dots, s_v, \dots, s_n)$ denote with $\bar{s}^v = (s_1, \dots, \neg s_v, \dots, s_n)$ the state which differs from s in the value of the component v . The transition relation is then defined by $s \rightarrow \bar{s}^v \iff K_v(\{u \in v^- \mid s_u = 1\}) \neq s_v$. This update schedule produces every possible trajectory emerging from a state by changing one value, thus the dynamics are non-deterministic and therefore more challenging in terms of computation and interpretation. More precisely, asynchronous dynamics can contain trajectories that are not predictive for biological behavior.

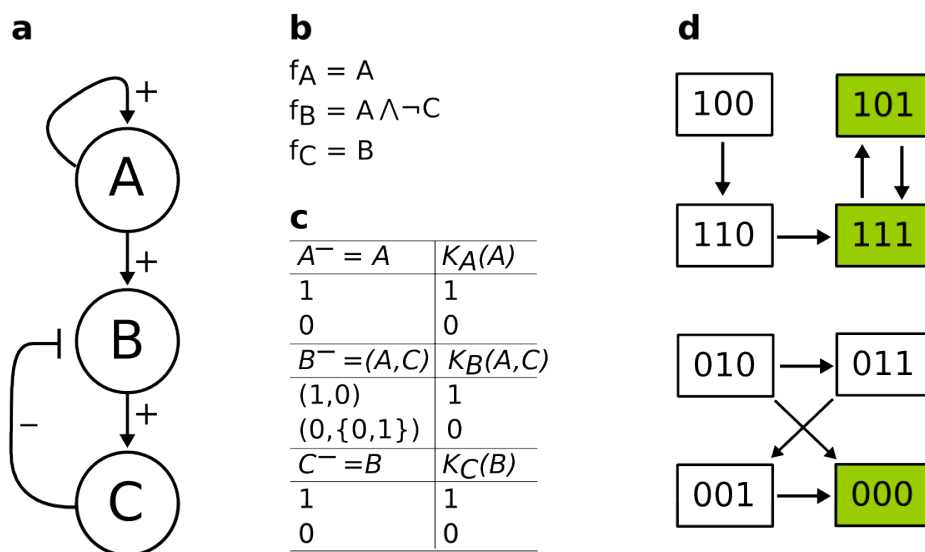


Fig. 2.1. Definition and visualization of the toy model. **a** IG of three components A, B, and C with edges and corresponding edges labels. **b** Logical functions for each component describe the regulatory context. **c** Alternative definition of the connected components is given by parametrization. **d** STG of the model splits in two distinct graphs depending on the state of component A. For $A=1$ all trajectories end in a cyclic attractor, where components B and C switch between states 0 and 1. For $A=0$ the system has a fixpoint with all components being inactive.

However, the dynamics often yield more realistic trajectories than synchronous update, since the assumption that every component changes its value at a different time point, even if the difference is very small, is biologically reasonable. For example, a system might contain signaling processes that are direct and usually fast as well as processes that require a translocation to a cell organelle, which takes more time, or even de novo synthesis of the protein via gene regulation processes are slow relative to direct activation. Since this information is often not given or not to the extend that a sequential update can be derived from this knowledge, every combination of sequences is produced. This issue is very important in this modeling

framework, since Boolean modeling is widely seen as formalism of choice when dealing with very large networks containing various processes.

2.3.2 Characteristics of dynamics

The complete dynamical behavior can be described as a state transition graph (STG) $\mathcal{R}(K) = (S, \rightarrow)$. This is again a directed graph, where the node set is given by the state space and the edges are determined by applying the logical equations according to the update strategy. The STG and its characteristics form the basis for the analysis of qualitative models.

The trajectories of the system are contained in the STG as paths, denoted as $Path = Path(S, \rightarrow)$ (notation adapted from [53]). Here, a path $\pi \in Path(S, \rightarrow)$ is a sequence (s_0, s_1, \dots) of $s_i \in S$ such that $s_i \rightarrow s_{i+1}$ for all $0 \leq i$. In case the path is finite, the sequence (s_0, s_1, \dots, s_k) is restricted to $0 \leq i < k$, where $k \in \mathbb{N}$ indicates the length of the path by the number of transitions. Then, the existence of a path from x to s is described as $x \rightsquigarrow s$. According to the square bracket notation from [4], the i th state in a path π is described as $\pi[i] := \pi_i$ and a fragment as $\pi[j..k] := (\pi_j, \dots, \pi_k)$. In the application to biological systems, these paths are compared to sequences of measurements of the system, where often the initial state or a set of initial states is known. Therefore, paths with a given initial state $s \in S$ or a set of initial states $IS \subseteq S$ are noted as

$$Path(s) := \{\pi \in Path \mid \pi[0] = s\}$$

$$Path(IS) := \{\pi \in Path \mid \pi[0] \in IS\}.$$

Regions in the state space that are highly connected are of special interest, because trajectories may stay for a long time in these regions, which means that these dynamics are likely to be an observable behavior of the system. A strongly connected component (SCC) is an inclusion-wise maximal subset $X \subseteq S$ that satisfies $x \rightsquigarrow y$ for all $x, y \in X$. This means that there exists a path between every state in this region of the STG. In the STG, a SCC that cannot be left by any trajectory is called an *attractor*. In case this set consists of a single state, we call it a *fixpoint* or steady state, otherwise a *cyclic attractor*. Each attractor separates the STG into basins of attraction, which contain all states that evolve into the respective attractor.

An example of a STG is given in Figure 2.1 d where two graphs are shown that describe the two basins of attraction. These basins are two distinct sets of states, since the component A is an input node and the state of A governs the behavior of

the system. Therefore it cannot change its state and for $A=1$ the attractor (marked green) is a cyclic attractor and for $A=0$ a fixpoint.

2.4 Model pool

When building a model, often one has to deal with sparse information about the biological system. For example, if the regulation of multiple genes is of interest, but the respective information about each individual gene is only available from different cell types or even species. In these cases, the logical equations or even the presence of a regulation is uncertain. Here, we address this problem by including this information into the modeling procedure by extending the set of possible edge labels to $\{+, -, \neg+, \neg-\}$. Here, the labels $\{+, -\}$ are assigned to edges of known interaction from literature either with an activating or inhibiting effect, respectively. These interactions are called *observable* meaning that the edge is present in every model, whereas edges that are not present are also called *not observable*. The labels $\{\neg+, \neg-\}$ are assigned to uncertain edges, which are controversial in the literature or hypothesized to be present. Here, $\neg+$ means that the edge is not activating, i.e. it is either not observable or inhibiting, and $\neg-$ means that the edge is not inhibiting, i.e. it is either not observable or activating.

For defining the model pool, every solution of K_v for the edge labeling is calculated, s.t. $l(u, v)$ evaluates to *true* for each $u \in v^-$. More specific, we say that K_v is a solution of the edge labeling *iff*

$$l = \begin{cases} + : & \forall \omega \subseteq v^- : K_v(\omega) \geq K_v(\omega - \{u\}) \wedge \\ & \exists \omega' \subseteq v^- : K_v(\omega') > K_v(\omega' - \{u\}), \\ - : & \forall \omega \subseteq v^- : K_v(\omega) \leq K_v(\omega - \{u\}) \wedge \\ & \exists \omega' \subseteq v^- : K_v(\omega') < K_v(\omega' - \{u\}), \\ \neg+ : & \forall \omega \subseteq v^- : K_v(\omega) \leq K_v(\omega - \{u\}), \\ \neg- : & \forall \omega \subseteq v^- : K_v(\omega) \geq K_v(\omega - \{u\}). \end{cases}$$

The parametrization $K = (K_v)_{v \in V}$ is a solution to l if K_v is a solution to l for each $v \in V$ [98].

In case there is information about the regulatory mechanism of a component available, we want to define an update function directly. This logical equation f is required to be consistent with the edge labels. The set of all K that are solution to l and f is called the *model pool*, denoted $\mathcal{K}(V, E, l, f)$ where each model contains

one unique parameter for every component. Thus, the model pool arises from the uncertain wiring and unknown regulatory mechanisms of a system.

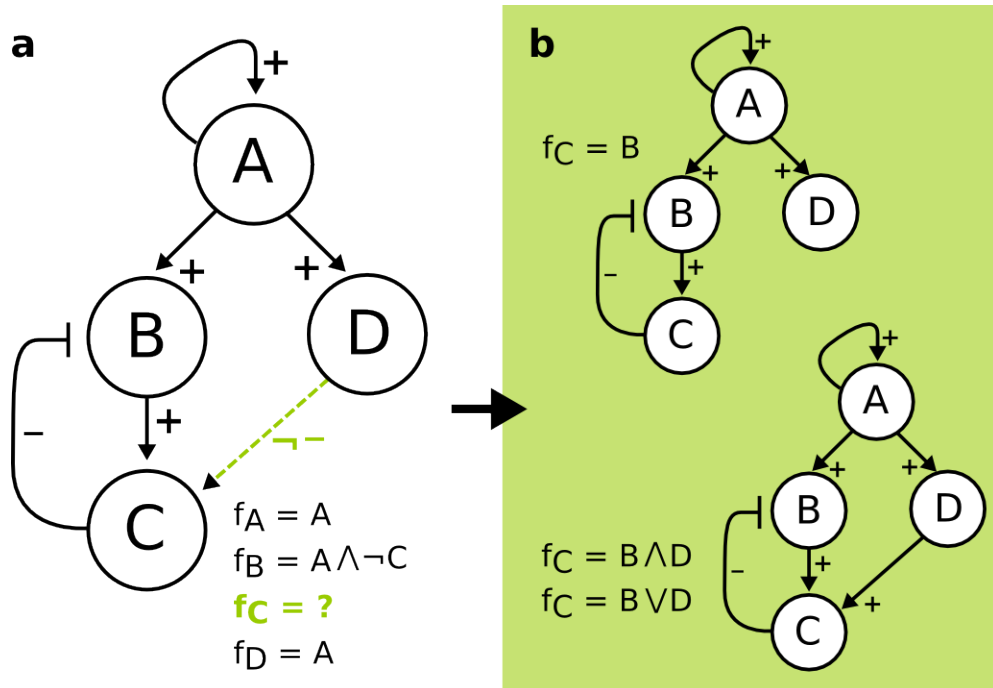


Fig. 2.2. Building the model pool for expanded toy model. **a** IG of the system with additional component D is activated by A and an optional edge from D to C is labeled as not inhibiting. The logical equations can be defined for components A, B and D, but not for C. **b** Two different topologies result from the graph, where in the bottom graph edge between D and C is present with two different logical equations possible, namely an AND or OR gate. In the upper graph the model does not contain the edge from D to C, thus B is the only regulator of C. Finally, the resulting model pool contains 3 models.

2.5 Data processing

In order to incorporate experimental data into the mathematical analysis, it needs to be processed depending on the type of data and the technology used for measurement. Often background noise needs to be filtered, big data sets require normalization and statistical evaluation before applying them to analysis methods. Since all data used in this thesis was already processed, we do not address this part of data processing any further. Any preprocessing we applied to data in our case studies will be explained within the case study later. However, in order to compare experimental data to discrete trajectories, we do need to discretize this data to match the logical formalism. Since we are only interested in basic dynamics such as finding a trajectory where a component switches on/off or oscillates, the discretization needs to preserve the information we are interested in for our analysis.

2.5.1 Discretization

The discretization process transforms quantitative into qualitative data, e.g. protein concentrations are assigned to a finite number of intervals resulting in distinct partitions of the continuous information [29]. Although the method comes with a loss of information especially for quantitative data, it also provides advantages for inferring knowledge from data. Discrete values are easier to use since the data is less complex and therefore learning algorithms can process this information faster and more efficiently [29]. Moreover, the amount of noise in the biological data can be reduced by discretization, as shown by Dimitrova et al., where time-series data was more robust to noise when compared to continuous values [25]. Finally, the interpretation of discrete values is often more intuitive, since biological experiments often have discrete characteristics like knockout of genes or stimulation of a receptor.

When discretizing data, an important decision is to choose the levels of discretization. That is, how many meaningful levels can be assigned to a functionality that corresponds to different states in the model. In the simplest case, the binary discretization $\{0, 1\}$ is used, where '1' represents an upregulation or activation and '0' represents a downregulation or inhibition. For discretizing data points into two states, a threshold needs to be defined. Here, different methods can be applied to find a threshold, such as mean or median [25].

Using a threshold for discretization causes every component to switch from 0 to 1 (or vice versa), no matter how small the variation in the activity is. Thereby, minor fluctuations of an active component can be discretized to an oscillatory behavior. To overcome this issue, an alternative widely used scheme is the ternary discretization, where three levels are considered $\{-1, 0, 1\}$ describing downregulation, no change and upregulation, respectively. In this scheme, an upper and lower threshold are set to define a region that is considered to not functionally show a specific activity. A common application is calculating the fold-change (fc) of measurements and defining a threshold of $fc = 2$ and $fc = -0.5$, which means that the activity needs to be doubled or halved to be up- and downregulated, respectively.

In theory it is possible to define an arbitrary high number of levels, called multivalue discretization. In case a component has a certain functionality at a low concentration and gains an additional functionality at a high concentration, this can be captured by assigning $\{0, 1, 2\}$ for inactive, low and high concentration, respectively. Again, a threshold between every level needs to be defined. With an increasing number

of levels the complexity of the data for analysis and interpretation increases, thus the choice of discretization is always a trade-off between level of detail and cost of complexity [29].

In this work, both binary and ternary discretization is used depending on the biological system that is modeled and the available experimental data. Specifically, we faced two different kinds of data sets, quantitative data having continuous measurement values and qualitative data. For quantitative data sets, the former procedure of discretizing by calculating thresholds is applied. For qualitative data sets such as western blots, only visual information is available. Here, measurements are usually interpreted relative to a control measurement. If a measurement is judged as ambiguous, it is excluded from the study.

Finally, the last processing step is assigning the discretized values to model components. More specific, we investigate the behavior of signaling processes, where kinases activate their target by phosphorylation. Thus, we define the presence of target phosphorylation as readout of the activity of a component, rather than the phosphorylation of the component itself.

2.5.2 Temporal logic

After discretizing the data to logical states, we also want to include information about transitions between states from the data to the model. In order to compare transitions in the STG with data, the measurements need to be interpreted in the STG by encoding this behavior as temporal logics. In general, temporal logics describe an ordering or a sequence of events in time with two different concepts: *linear time logic* (LTL) and *computation tree logic* (CTL), which describe deterministic and non-deterministic sequences of events, respectively. For our analysis, two different tools were employed. Either we explicitly defined CTL queries from our discretized data or we used a software which provides an interface to enter the data and then builds the temporal logic autonomously. Note that LTL formulas are incomparable to CTL formulas and since we do not formulate LTL queries ourselves in this work, no formal definition is given. However, most of the dynamics we consider can be described by an equivalent LTL formula.

Computational tree logic Computation tree logic is a branching time logic introduced by Clarke [17], designed for systems that at each point have many possible futures. Here, beginning in any state $s \in S$ evolving in time is represented as alternative paths in trees and sub-trees of possible future states. CTL formulas

consist of atomic propositions (AP), which are combined using Boolean operators and temporal operators. Along with definitions in [53], the syntax of CTL formulas is divided into a state formula and path formula. The state formula ψ is defined over a set of AP by the following grammar:

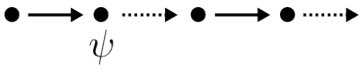



$$\psi ::= true \mid a \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \mathbf{A}\varphi \mid \mathbf{E}\varphi$$

where $a \in AP$ and φ is a CTL path formula formed according to the grammar:

$$\varphi ::= \mathbf{X}\psi \mid \mathbf{F}\psi \mid \mathbf{G}\psi \mid \psi_1 \mathbf{U}\psi_2$$

where ψ, ψ_1 and ψ_2 are CTL state formulas.

There are two kinds of temporal operators, the first one quantifies over paths $\{\mathbf{A}, \mathbf{E}\}$, where \mathbf{A} means the formula is valid, if φ holds on all paths starting from the current state. The operator \mathbf{E} means the formula is valid, if there exists at least one path starting from the current state where φ holds. These operators can be interpreted as the strictness for applying the path formula. The operators $\{\mathbf{X}, \mathbf{F}, \mathbf{G}, \mathbf{U}\}$ are path-specific with the following semantics.

- | | |
|--|--|
| X: Next: ψ has to hold at the next state. |  |
| G: Globally: ψ has to hold for the entire subsequent path including the current state. |  |
| F: Finally: ψ has to hold somewhere on the subsequent path. |  |
| U: Until: ψ_1 has to hold at least until ψ_2 , which holds at the current or a future position. |  |

Usually there are two different kinds of temporal behavior observed in experiments: time-series measurements or steady state measurements. In logical modeling, we interpret time-series as trajectories in the STG and steady states as fixpoints of the system. Using the path operators, time-series are encoded as a sequence of states, where the ordering of events is preserved. In Figure 2.3 an example for the data processing of time-series data is given. Here the CTL formulas consist of series of states that should exist at some point in the future. For both formulas a state is not required to hold until the next state is reached and also we do not require the next measurement to hold at the next state. Therefore, $\mathbf{EF}(\psi_1 \ \& \ \mathbf{EF}(\psi_2 \ \& \ \dots \ \mathbf{EF}(\psi_n)))$ is the structure we employ for encoding time-series measurements. Note that in the CTL formulas we use the symbol $\&$ for \wedge .

For attractors, the path operator \mathbf{G} can be used if a longterm behavior of a component is known. For example, if we would have the information that components B, C, and D in the toy example stay inactive when A is off, we could include this in our CTL as $\mathbf{EF}(\mathbf{AG}(A = 0 \& B = 0 \& C = 0 \& D = 0))$. This formula describes an attractor, since all components are specified to stay at a certain value. In case only a subset of the components of the system are assigned to a state all other component would be allowed to change, which would be a cyclic attractor.

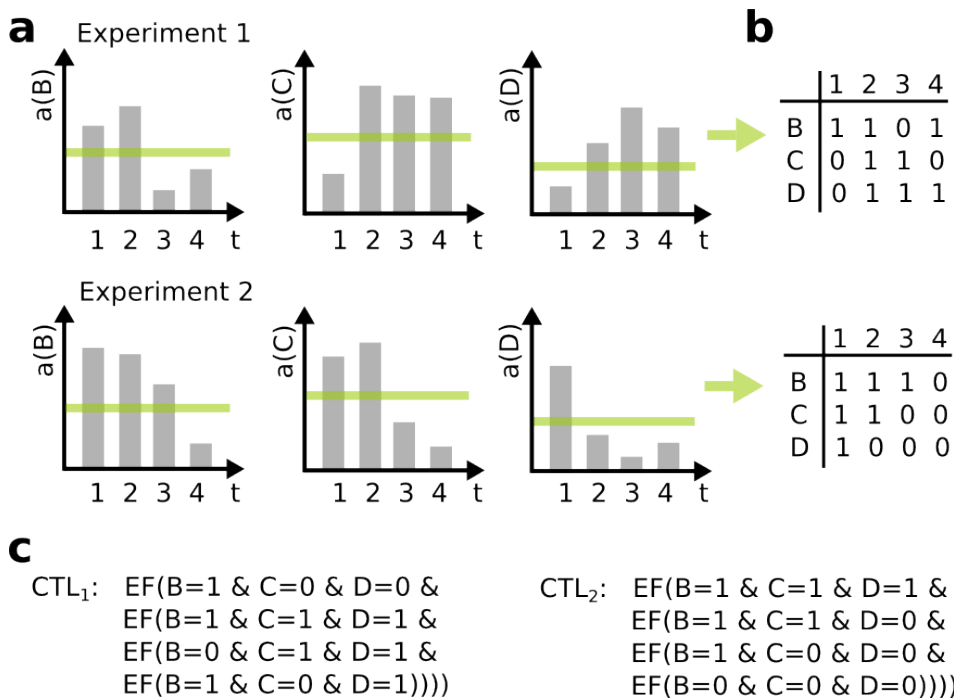


Fig. 2.3. Discretization and formal encoding of toy data. **a** Two experiments with time series measurements over four time points observed the activity of components B, C and D denoted as a(B) etc. The green line represents the threshold for binary discretization. **b** The tables show the discretized data for each component and each time point. **c** CTL formulas derived from tables **b** are shown for each experiment.

2.6 Model checking

Model checking is a formal method from computer science that has been shown to be a powerful tool for analyzing biological problems [71, 6, 11]. Different kinds of temporal information from the biological systems or data is checked for agreement with a mathematical model. After encoding the observed information as temporal logics, powerful algorithms are used to compare the model with the formula.

Given a model and its STG, model checking is able to decide whether or not a temporal logic specification is satisfied by the model dynamics. For this purpose, a Kripke structure is used, which describes a transition system (TS) based on a finite set

of states, a set of initial states, a transition relation, a set of atomic propositions, and a labeling function. For an STG, the $TS = (S, \rightarrow, IS, L)$ with L being the labeling function from states to atomic propositions (for more details see [4]). For a path in the TS, the satisfaction relation $s \models \psi$ which defines whether a state $s \in S$ satisfies a CTL state formula ψ is given by

$$\begin{aligned}
s &\models true \\
s &\models a \quad \text{iff } a \in L(s) \\
s &\models \neg\psi \quad \text{iff } \text{not } s \models \psi \\
s &\models \psi_1 \wedge \psi_2 \quad \text{iff } s \models \psi_1 \text{ and } s \models \psi_2 \\
s &\models \mathbf{E}\varphi \quad \text{iff } \exists \pi \in Path(s) : \pi \models \varphi \\
s &\models \mathbf{A}\varphi \quad \text{iff } \forall \pi \in Path(s) : \pi \models \varphi.
\end{aligned}$$

If we extend the satisfaction relation from paths to transition system TS, we want to quantify the satisfaction by either having one $s \in IS \supseteq S$ or every initial state as condition [53]. Then $TS \models \psi$ iff

$$\begin{aligned}
\exists s \in IS : s \models \psi \quad &\text{called } ForSome \\
\forall s \in IS : s \models \psi \quad &\text{called } ForAll.
\end{aligned}$$

The satisfaction relation for $\pi \models \psi$ for paths $\pi \in Path(s)$ describing the CTL path formula is given by

$$\begin{aligned}
\pi &\models \mathbf{X}\psi \quad \text{iff } \pi[1] \models \psi \\
\pi &\models \mathbf{F}\psi \quad \text{iff } \exists 0 \leq i : \pi[i] \models \psi \\
\pi &\models \mathbf{G}\psi \quad \text{iff } \forall 0 \leq i : \pi[i] \models \psi \\
\pi &\models \psi \mathbf{U} \psi \quad \text{iff } \exists 0 \leq j : \sigma[j] \models \psi_2 \text{ and } \forall 0 \leq i < j : \sigma[i] \models \psi_1.
\end{aligned}$$

For our toy example, two CTL formulas from Figure 2.3 c are checked on the STGs of the three models from Figure 2.2. In Figure 2.4 this process is visualized by marking the matched states. Here only data for components B, C, and D is given, thus the CTL formulas are checked for two different initial states. Applying the formulas with the quantification option *ForSome* shows that experiment 1 must have been observed with active component A, since only that part of the STG was able to match CTL_1 for some models, vice versa for experiment 2. Applying the formulas with quantification option *ForAll* would lead to no match for both experiments.

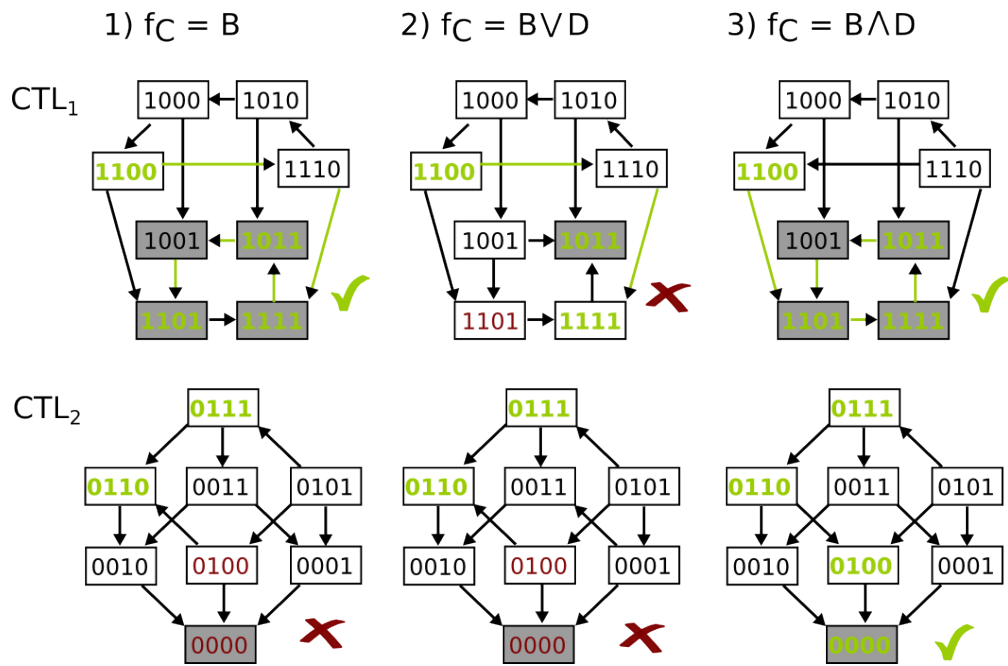


Fig. 2.4. Model checking for CTL_1 and CTL_2 reveals only model (3) is valid for both data sets. The STGs of all three models from the model pool (see Fig. 2.2) are shown represented by their logical formula of component C. Each STG is split into two distinct graphs since component A cannot change its value. Each STG was checked for both CTLs, but only some dynamics with active A were able to match CTL_1 and some dynamics with inactive A matched CTL_2 . Moreover, only models (1) and (3) are in agreement with CTL_1 and model (3) with CTL_2 , where each matched state is marked green and models matching all four states are marked with checkmark.

2.7 Software

In the field of systems biology, there are many tools for mathematical modeling of biological systems. For example, CellNetAnalyzer provides a graphical user interface and various computational methods and algorithms for exploring structural and functional properties of metabolic, signaling, and regulatory networks [52]. It incorporates methods for functional network analysis like characterizing functional states, detecting functional dependencies, identifying intervention strategies, or giving qualitative predictions on the effects of perturbations. A specialized tool for logical modeling is GINsim, which allows to build a logical model, build and visualize the STG for both synchronous and asynchronous update and simulates knock-outs in components [72]. However, there is no specific tool for building pools of logical models and testing them for data. In our group, two different tools were developed to allow for such an analysis: TomClass and Tremppi. The tools and applications were published [53, 90, 92, 98]. In the following a short description of the tools is given.

2.7.1 TomClass

The first software I used for my analysis is a Python based model checking tool called TomClass developed by Hannes Klärner, which builds and analyzes model pools [53]. The tool can be divided into two parts: model instantiation as well as annotation and model pool analysis.

Model instantiation and annotation The first step is to define the topology of the system. As input all prior knowledge on components $v \in V$, including their maximal activities and names, and their interactions $e \in E$ is entered, together with a threshold function θ . The prior information on the regulations of the components are entered either as logical equations for components that have known parameter values or as edge labels for uncertain regulatory contexts. For enumerating all models arising from the input, the expressions are translated into a constraint satisfaction problem (CSP) and solved using the CSP solver Python-constraint, see [53].

The resulting model pool is stored by the database engine SQLite. The database consists of a single table and each model of the pool is stored in a single row. Further information is added in dedicated columns which is done in the annotation step. During the annotation phase models are tested typically by model checking or other properties that require exploring the state transition graph [53]. According to Klärner, a test is any algorithm that computes a label for a model. Here, annotation scripts require a property name under which the annotation is stored as a column in the database and the model pool that is tested. There is also the option to test only a subset of models in case testing the full pool would be computationally challenging, but this option is not used in our analysis.

For performing model checking on a model pool, TomClass employs NuSMV [15, 14], which is based on Symbolic Model Verifier (SMV) that was developed by E. M. Clarke et al. [16]. NuSMV is able to efficiently verify LTL and CTL specifications for finite transition systems. A model is first translated into the NuSMV language and then passed to the software together with a LTL or CTL specification for verification [53].

There are three annotation options available in the script: *annotate_reachability*, *annotate_attractors* and *annotate_crosstalk*. Here, reachability means testing whether the transition system of a model is able to satisfy a CTL or LTL formula and its constraints, which is called property. For this aim, the property is defined in the script with the following parameters:

- `pname`: is the name for the label of the property for annotation in the SQL database.
- ψ : is the CTL formula as described in Section 2.5.
- `Delta`: describes the delta constraint, which defines whether a component is required to decrease or increase its value or not. `Delta=1` means the component must change its value in the future and `Delta=0` means that no change is possible, i.e. we are in a steady state.
- `v=b`: where $v \in V, b \in \mathbb{B}$ states that value of a component v is set to b .
- `Initial state`: is a list of Boolean constraints on the values of the components. A state is considered initial, if all the constraints are satisfied.
- `Verification_type`: can opt between the satisfaction conditions `ForAll` and `ForSome`.
- `Fixed component`: constrains the listed components to the assigned values for the whole path. This parameter allows for modeling knock-outs and stimuli.

There is also an option to passing a matrix of values together with a vector of the component names to the program, which then generates a time-series CTL formula automatically. The first row of measurements, meaning the first time point, is used as initial state and the last time point can be set as fixpoint by another parameter, i.e. `Fixpoint` can be set to true or false.

The algorithm `annotate_attractors` requires a declaration of the input nodes and their initial states. Then, the attractors for every combination of the given initial states are calculated and added to the database as properties, where every combination results in a new column. For specifying the attractor, all components that are stable are listed with their respective value. Thus, if every component of the system is listed the input results in a fixpoint.

For `annotate_crosstalk` or in general optional edges, all edges of interest are listed in a vector. These edges are then tested in every model of the pool to be functional or not. This information is added to the database as a new column having 1 for an observable edge and 0 for non-observable.

Model analysis In the tool, the model pool is analyzed by classification, where the models are grouped into classes according to the annotated properties in the

database by the algorithm *analyse_classes*. Either all properties can be used as a classifier or a list of properties are defined in the parameter `Classes`. For example, we want to group the models according to their status of an annotated CTL formula. Also we can restrict the pool to a subpool using the parameter `Restriction`, where we can select models for their property, e.g. only including all models that satisfy a CTL formula or carry an optional edge. Mathematically, the analysis finds subsets of models that have a non-empty intersection and computes the cardinalities of these sets [53]. For this aim, an SQL query is generated using statements of the form

```
SELECT DISTINCT Classes FROM models WHERE Restriction
```

where `SELECT DISTINCT` computes all combinations of labels, i.e. subsets, of the selected `Classes` in the database `models`, possibly restricted using `WHERE`. Additionally, `COUNT` is used to determine the cardinality of each subset, i.e. the number of models in a class later denoted as size of a class. It is possible that classes are empty if there exists no model in the pool with a particular label combination. The script prints only all non-empty classes and their cardinalities to a CSV file [53].

For the analysis of the toy example, all properties were selected, which is the optional edge from D to C and the CTL formulas CTL_1 and CTL_2 . The resulting classification in Table 2.1 shows that every model is grouped in a separate class, since the labeling of the properties is different for each model. Thus, the size of every class is one, which is 33,3% of models per class of the pool. The labels of the classes agree with the observation in Figure 2.4, showing that the model without the optional edge from D to C cannot satisfy CTL_1 and only one of the two possible models containing the edge is able to satisfy CTL_1 and CTL_2 . However, from the classification it is not possible to determine which logical function is necessary to satisfy the formula.

Tab. 2.1. Output table of *analyse_classes* in TomClass for classifying the model pool of the toy example for the optional edge from D to C and the CTL formula CTL_1 and CTL_2 .

Properties			Size	
$D \rightarrow C$	CTL_1	CTL_2		
0	1	0	1	33.3%
1	0	0	1	33.3%
1	1	1	1	33.3%

This issue can be solved by another algorithm implemented in TomClass called *annotate_controls*, which calculates and returns logical formulas for every component in the pool. Applying this annotation to the full pool gives a list of every possible

logical equation for each component and a count for how many models carry this equation across the pool. However, this algorithm also uses the parameter `Restriction`, thus the composition of regulations can be determined to a resolution of a single class. In case a class contains a single model, the logical equations of that model are given. Accordingly, for a `Restriction: CTL1 = 1 and CTL2 = 1` the output is: $CompD \wedge CompB$. This result matches the observation in Figure 2.4.

2.7.2 Tremppi

The second software I used for building and analyzing model pools is called Tremppi (Toolkit for Reverse Engineering of Molecular Pathways via Parameter Identification) and was developed by Adam Streck [90]. The tool was built with the goal to provide a platform that allows to incorporate as much information as possible about the biological system and use this information to optimize the modeling and analysis process. The workflow (Fig. 2.5) is similar to TomClass. First, all possible models are enumerated according to the constraints given and stored in a SQL database. Then, these models are evaluated and labeled for properties like dynamical behavior. Next, a subset of these models can be selected, analyzed using various tools and compared to other selections. Finally, the selection or analysis can then be refined based on the newly gained knowledge [90]. Each of these step will be explained in the following.



Fig. 2.5. Workflow implemented in Tremppi (picture taken from [90]), with a five step process for enumerating, labeling, selecting, analyzing and comparing models in the model pool.

Enumerate In contrast to TomClass, the prior knowledge on the model topology and logical equations is not implemented as a script but through a user interface. The components and interactions are build as a graph and the edge labels are selected from a list, which consists of optional and fixed labels. For our applications we used a subset of possible labels, for more information see [90]. Fixed labels are `{Activating Only, Inhibiting Only}`, which means that the edge is always present and activating or inhibiting, respectively. Thus, they are the same for every model in the resulting pool. Optional edges are `{Not Inhibiting, Not Activating, Observable}`, which means that they are optional, but if they are present they are activating, inhibiting or any, respectively. These edges lead to differences in the topology of the models in the pool. Moreover, constraints on

the regulatory context of a component can be added. Once the regulatory graph $\mathcal{R} = (V, E, l)$ and possibly constraints are entered, all models fitting the network are enumerated building the model pool \mathcal{K} and stored in an SQL-database.

Label In order to further restrict the model pool and to apply biological information to the pool, labels can be added to the models in the database (not to confuse with edge labels). Here is a list of possible labels adapted from the website¹:

- $K_v(v^-) \in [0, Max(v)]$ is the parameter value for the component v in its predecessors v^- . The symbol $Max(v)$ denotes the maximum activity of v .
- $bias(v) \in [0.0, 1.0]$ denotes how much the component v has a tendency towards the lower or higher activity levels. It calculates an average activity value of a component across all parametrizations, where high values indicate a high influence of that component.
- $impact(u, t, v) \in [-1.0, 1.0]$ denotes how prominent the regulation (u, t, v) is. Values close to -1.0 denote a strong inhibition, the ones close to 0.0 denote weak effect, and the ones close to 1.0 denote a strong activation. More details are given in the **Analyze** paragraph.
- $sign(e) \in \{0, +, -, 1\}$ is the sign of the edge. The value 0 denotes no effect, the value + activation, the value - inhibition, and the value 1 activation and inhibition at once.
- $cost(property) \in [0, !0]$ denotes how many simulation steps it takes to satisfy the property. Low numbers indicate a short path and therefore easier agreement, however the special value 0 means it is not possible at all. For our applications, we only consider the options satisfiable (!0) and non-satisfiable (0).

Here, `cost` is a label calculated for a property, which is an encoded observation that is tested on the model pool. Since we are working with biological systems, these properties are usually experimental data which was discretized as described before. In contrast to TomClass, Tremppi does not use CTL formulas for model checking, but Büchi Automata. In general, Büchi Automata based model checking is used for properties described using the Linear Temporal Logic (LTL), but, as described by Streck [90], the tool is implemented to build properties as the more expressive Büchi Automata directly.

¹<http://dibimath.github.io/TREMPPI/> version TREMPPI 1.1.0, date 05.9.2016

In Tremppi, each data set is translated into a property $P = (\vec{M}, \vec{D}, End, Exp)$ where \vec{M} is a sequence of measurements, \vec{D} is a sequence of delta constraints, End is an ending, and Exp is an experiment [90]. In the following list, these elements are described in short (adapted from and for more details see [90]).

- **Sequence of measurements:** A single discretized measurement is described by a vector $M = \prod_{v \in V} m_v$ where $m_v \subseteq [0, Max(v)]$ a product of threshold values for a subset of components. These thresholds correspond to states of the components in the transition system or a set of states, if only a subset of components is measured. A sequence of measurements is then described by a vector $\vec{M} = (M^1, \dots, M^n)$ for some $n \in \mathbb{N}_1$. A path in the transition system matches the measurements, if there exists a sequence of states that matches the measurements in the given order.
- **Sequence of delta constraints:** With this constraint we can restrict the behavior of the components between the measurements, e.g. to enforce monotonicity in the sequence. Thus, an additional constraint assigned to each measurement is defined called *component delta* $D \in \prod_{v \in V} \{up, down, stay, none\}$ where *up* means the component can not decrease its value, *down* means the component can not increase its value, *stay* means the component can not change its value, and *none* means the component is not constrained.
- **Ending:** For each measurements sequence \vec{M} the ending is defined as a logical variable $End \in \{open, stable, cyclic\}$. Here, *open* means that there is no restriction on the dynamics after the last measurement. The option *stable* means that we assume the last measurement to be a steady state. The option *cyclic* encodes that after the last measurement the system returns to the first measurement in the sequence and thereby forming a cyclic attractor.
- **Experimental setup:** Often experiments include manipulations of the system which affect the model, e.g. often components are inhibited which affects their influence on other components. These manipulations can either restrict the state space or change the parametrization of the model. Then, the experimental setup for component v is denoted as $Exp_v \in \{[i, j] \mid 0 \leq i, Max(v) \geq j\}$ and the experimental setup of the whole network is $Exp = (Exp_v)_{v \in V}$ [90].

Select For every assigned label or set of labels, a selection can be created in Tremppi. These selections are used to filter the model pool for those models that are in agreement with the labels in the selection. Depending on the label, different

constraints for selecting labels can be entered in the software, from intervals of floating numbers to logical expressions. Here, I only used the restriction that can be applied to all labels, which is denoting !0 for the label being fulfilled and 0 for the label not being fulfilled. A selection is the conjunction of all labels, thus models have to meet all labels within one selection to be chosen. However, each selection itself is independent and applied in disjunction, which means that a model rejected by one selection is still checked for other selections.

Analyze Each selection creates a model pool, which can be analyzed with different tools, the so-called reports in Tremppi. There are two reports to summarize different characteristics across the pool: the qualitative and the quantitative report. The quantitative report provides the basic quantitative information about the selected model pool by giving a summary about numerical values in a table. In particular, there are 5 measures given: `Label` shows the label and its parameters for this row, `Count` shows how often the label has a non-zero value, `Min` and `Max` give the minimal and maximal value in the selection, and `Mean` gives the mean of the selection. The qualitative report gives a summary on all assigned labels and its parameters with the number of distinct values with the symbol `#` and the proportion of each value in the pool as `Elements`.

The report used throughout this work is called `regulations` report, which is a graph-based report which re-creates the network made in the editor. It visualizes the statistical analysis of the model pool and was introduced in [98]. First, the correlations between the states of the system and kinetic parameters of the individual component are computed to evaluate the effect of regulations. Note that the parametrization function describes a causal relationship—the dynamical behavior of a component is implied by the state of the system. Thus, for each pair $(u, v) \in E$ and a parametrization K we therefore compute the impact of u on v as the correlation between the current value of u and K_v . Formally, for each $K \in \mathcal{K}$ the impact function $imp_K : E \rightarrow [-1, 1] \subset \mathbb{R}$ is defined as $imp_K(u, v) = corr((s_u)_{\{s \in S\}}, (K_v(s))_{\{s \in S\}})$ where $corr$ is the Pearson product-moment correlation coefficient. This notion is extended to parametrization sets by employing the mean. Formally, an extended impact function $imp_{\mathcal{K}} : E \rightarrow [-1, 1]$ as $imp_{\mathcal{K}}(u, v) = \frac{\sum_{K \in \mathcal{K}} imp_K(u, v)}{|\mathcal{K}|}$ is created. Note that $imp_K(u, v) = 0$ is equivalent with $l(u, v) = (\neg+) \wedge (\neg-)$, meaning the edge is non-functional. However, in case an edge is labeled with $l(u, v) = (+ \vee -)$, both negative and positive edges can appear within one pool and theoretically could lead to $imp_K(u, v) = 0$ as well. In the case studies, I do not include edges with ambiguous labels, thus the issue does not affect my analysis. Another issue of the impact value is that it is always split upon all predecessors and therefore results in lower impact

values for edges that influence components with many incoming edges. Thus, also low impact values must be considered in the analysis.

Lastly, the second measure for the regulations report is how often an edge is functional in the resulting set, which is described by the frequency function $freq_{\mathcal{K}} : E \rightarrow [0, 1]$ defined as $freq_{\mathcal{K}}(u, v) = \frac{|\{K \in \mathcal{K} | imp_{\mathcal{K}}(u, v) \neq 0\}|}{|\mathcal{K}|}$. In the report, the mean impact in the selection is projected to the color of the regulation, while the width of a regulation is obtained from its frequency (i.e. how often it takes on a non-zero value). Moreover, a comparison between two regulations reports of the same system can be created, where the statistical measures are subtracted from the reference. Again the impact is illustrated using colors and the frequency as thickness of lines, where non-positive frequency can occur as dashed lines and the 0 value is displayed as dotted lines. Moreover, one can opt for the relative representation of the statistics, where the values are normalized in the current report to the boundaries.

There are three more reports available, which are not used in this study. The correlations report visualizes the values of the bias label of a component and the correlation between bias. The group report creates mutually disjoint sets of parametrizations that match on selected features, thus classifies groups or models with common labels. The witness report provides a statistic on paths in the transition system called witnesses. It visualizes those with the minimal cost, for all the properties selected at once. For more details on the reports see [90].

Compare In the last step of Tremppi's workflow, it is possible to compare the analysis of different selections. Within each report, the difference between two selections can be calculated in order to evaluate the effect of the different selections on the model pool. In general, the most suited report strongly depends on the pool structure and size.

Even after incorporating numerous properties, the resulting pool \mathcal{K} may be too large for manual analysis. In that case the regulations report can capture the nature of the selection. Here, a possible comparison would be the unfiltered pool and the selected pool to visualize the impact of the selected labels on the pool. Another example is to use the quantitative report to compare the parametrizations of two selections.

Toolbox for evaluating uncertainty in biological systems

Mathematical methods are artificial constructs used to help understanding biological processes. In order to receive meaningful results from a modeling study, the biology needs to be transferred into mathematics and the results need to be interpreted from a biological perspective, which is not straight-forward. Here, we address this task of incorporating biological information into the formalism presented in Chapter 2 in a four-step workflow: system initialization, objective formalization and adaptation, data formalization, and pool analysis.

When modeling a biological system, there are very different requirements and issues to address depending on the structure and the aim of the study. The workflow in Figure 3.1 gives a general approach for building logical models for problems I came across in my studies as a toolbox of methods. At first, the process of bottom-up model building formalizes the biological phenomena into the prior knowledge network, which we call system initialization. Here, the regulatory graph and the logical equations are derived from literature information. Then, the objective formalization includes the aim of the study into the model setup, e.g. by adding extra components or changing the labels of edges. After generating the model pool, the top-down filtering process uses biological data that needs to be encoded into temporal logics, the data formalization. Finally, the pool analysis gives two options to examine the specific pool for new biological insight. Although the workflow was developed for signaling networks, the approach can be applied to any related modeling problem.

3.1 System initialization

The first level of incorporating biological information into the model is the model building process itself. Here, literature information is gathered and interpreted to build the prior knowledge network, which forms the basis for the analysis. Depending on the aim and available information of the system, very different models can be build even if the same system is examined and the same data is used. The model

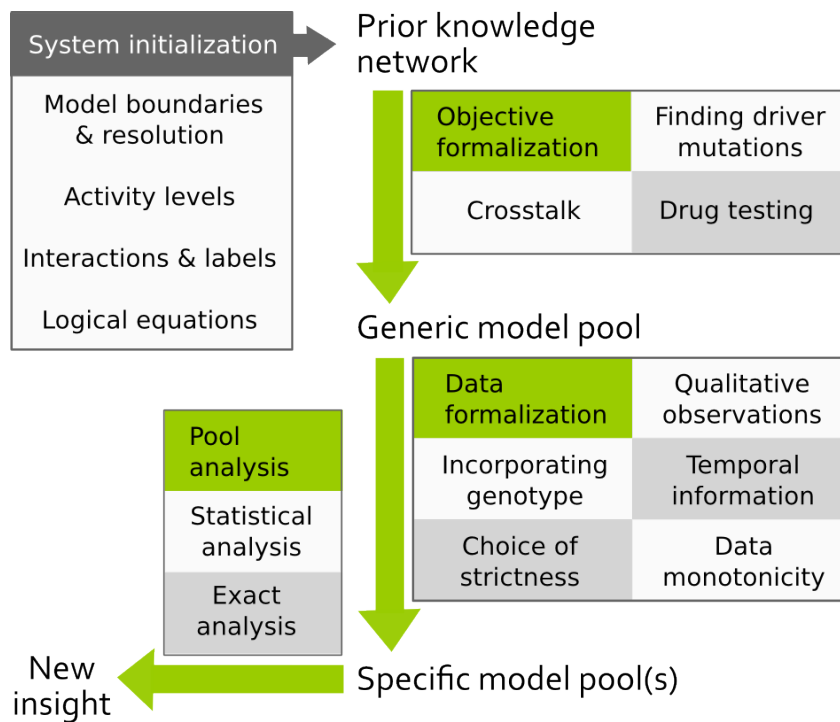


Fig. 3.1. Toolbox for evaluating uncertainty in biological systems as a workflow. For building the prior knowledge network and defining the uncertainties of the system, the system initialization and objective formalization is necessary. The filtering process from the generic model pool to the specific model pool requires data formalization and the interpretation of the final pool is done by pool analysis.

building process is decisive for the later outcome of the analysis and needs to be done with care, still this aspect of modeling is often underrated. Since there is no clear ruling on how model building should be done, I will present my approach here and compare it with the literature.

3.1.1 Model boundaries & resolution of entities

The first decision that needs to be made for model building is to define the scope of the model. Since it is only possible to model a fraction of the highly complex biological system, those elements that are necessary for understanding and representing the biological phenomena should be included. In the cell biological scale this means that signaling processes are usually obtained in the enclosed pathway [78], gene regulatory processes examine the transcription factors, genes and feedbacks [68], and metabolic processes are represented by enzymes, nutrients and metabolites involved [84]. However, these processes do not work in isolation of the cell, therefore the boundaries for the model need to be set with care and the interpretation of results need to be done in awareness of the biological context.

For my studies, I consider signaling pathways where the structure is often a receptor as input and a subsequent cascade of kinases, phosphatases and other proteins. Often a signaling pathway regulates one or more transcription factors and thereby genes and cellular reactions. I set the model boundary at the last measured component in the cascade (unless a component further downstream influences a feedback) and do not include transcription or cellular reactions such as apoptosis into models, since there are too many possible influences from outside the model boundaries.

There are many different ways to translate a system with its components and interactions to a model as defined in Section 2.1 depending on the focus, the available data and the aim of the study. Models can be built for various aims, e.g. to be biologically descriptive, to illustrate specific mechanisms of the system, or to be minimal to measures like number components or interactions [3].

To illustrate this issue, Figure 3.2 shows a toy example of a signaling process, where A is the receptor which phosphorylates B and D. Phosphorylated B acts as a transcription factor for gene c and causes the production of protein C. This protein C becomes phosphorylated by active (phosphorylated) D, which is then able to bind the unphosphorylated form of B and thereby prevents its activation by A. From a modeling perspective, there are essential and non-essential elements in this system. The receptor A is necessarily the input of the model, and B is important as activator of C and recipient of the negative feedback from C. Then C integrates the signals from B and D and triggers the negative feedback on B. These processes determine the behavior of the system, but nodes that simply receive and pass a signal are not essential. The component D is such a node, therefore if one aims to build the *minimal model* this component is left out, see Fig. 3.2 a. A chain of these nodes is called cascade and is often simplified in models without continuous time, but also in discrete time models they can be included, e.g. to observe delays. However, since we are not interested in such delays, we consider cascades as non-essential.

It is common that genes and their proteins are modeled in one component, but this simplification is a strong assumption. In the toy example, information about the influence of B on the gene c and of D on the protein C at two different phases, pre- and post-transcriptional, is lost in model Figure 3.2 a. Therefore, it can be beneficial to distinguish between these components to make it more descriptive and better resolve the temporal actions of B and D (see Fig. 3.2 b). Another possible step of refinement is to add a node for the bound components B and C as a complex. This step illustrates the biological mechanism by which the inhibition of C on B is working, shown in Figure 3.2 c. Then the complex inhibits B from activating c .

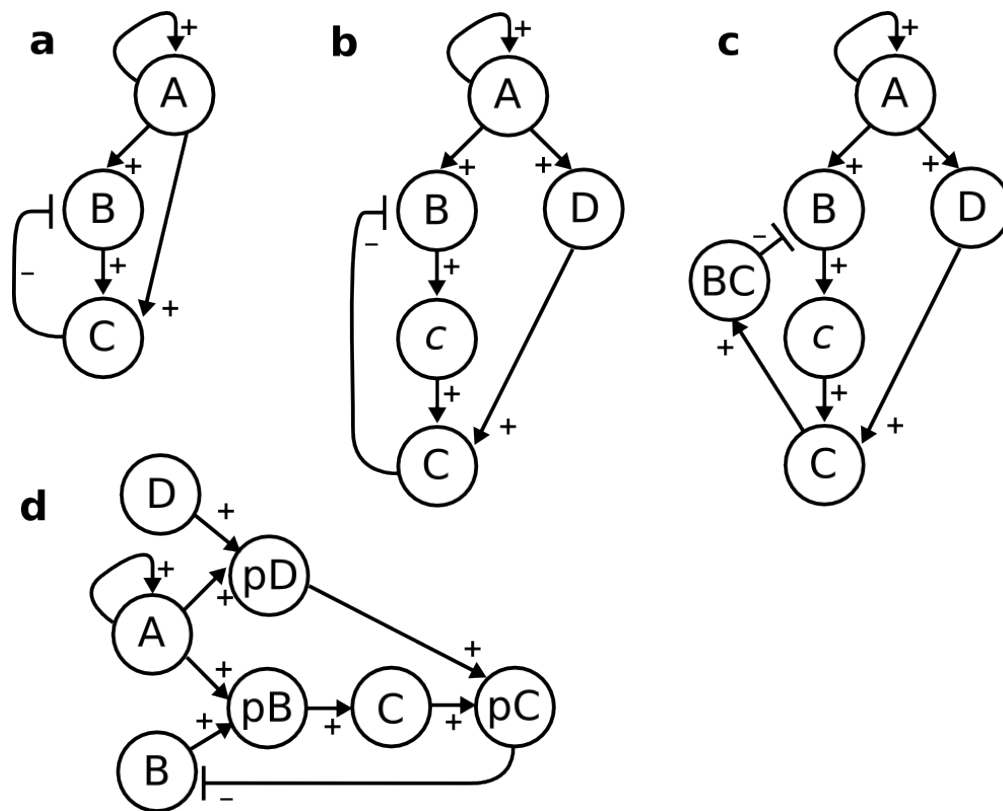


Fig. 3.2. Resolution of entities in the same toy system gives different models. **a** Original toy model build with lowest resolution. **b** Gene *c* is separated from its protein C. **c** The complex BC is modeled explicitly. **d** Proteins and phosphorylated proteins are modeled as distinct components.

However, the models in Figure 3.2 **a-c** do not illustrate whether an interaction acts on the phosphorylated or unphosphorylated form of the protein. Especially, when the modeler wants to emphasize that both forms can be present at the same time, each form is modeled as its own component. For the toy example, this results in at least three extra components (the receptor could activate itself by autophosphorylation leading to another component). In this model, the information about pC binding B and not pB is added (Fig. 3.2 **d**).

Finally, it is possible to combine these different resolution levels to receive very descriptive models, but every additional component increases the complexity of the system and its analysis. Here, the smallest model describing the toy example contains three components A, B, and C resulting in a state space of 2^3 , 8, states. The most comprehensive model, including the separation of *c* and C, the complex BC and adding the phosphorylated forms of the protein leads to a model with eight components and a state space 2^8 , 256, states. Thus the resolution of the entities and along with that the complexity of the model is a trade-off between being biologically descriptive and complexity in form of computational costs and interpretability.

I choose the resolution based on the functionality of a component and the data available. In the toy example, a distinction between gene c and its protein C were included if there were measurements of both the transcription, e.g. mRNA of c , and the protein activity, e.g. phosphorylation of C. Otherwise, c simply passes the activation from B to its successor C, thus deleting this node does not change the sequence of events between B and C. Adding a complex built from two other present components to the model, can aid understanding the biological mechanisms of a system. In the toy model, the inhibition of C on B by binding is not visible without the complex. However, even if there is data of the complex present, the existence could also be interpreted as active C, which leads to the same dynamics than with the complex. Also, splitting a component to its phosphorylated and unphosphorylated form is only applied, if both forms give distinct functions and both are directly or indirectly measurable.

As a result of choosing the model boundaries and the resolution for every entity within these bounds, the set of components V is defined.

3.1.2 Activity levels

Not only the resolution in terms of biological entities is an important step when building a model. Also, the decision of the level of abstraction in terms of the functionality of a single component itself, the activity levels, needs to be deduced from biology. As described in Section 2.1, logical modeling requires components to have discrete levels of activity. In case a component is only modeled for one functionality the Boolean description is sufficient, with 0 being inactive and 1 for expressing that functionality, which is the most common form of logical models. Otherwise a multilevel representation can be used to express that a component is e.g. not present (0), present (1), and active (2) or has a low, medium and high concentration. Of course, there is no limit in the number of levels, but similar to the number of components the complexity increases with the number of levels. Especially, the thresholds distinguishing the activity levels are additional parameters of the system that need to be determined from data.

In general, multilevel representation of a component is meaningful, if the effect is linear in the activity of the component. For example, in Figure 3.3 a the behavior of B in the diagram is linear, meaning that it activates D at a medium activity (1) and additionally acts on C for high activity (2). Then the oscillations caused by the negative feedback on B does not affect the regulation of D. However, if the measurement of B would be an overlay of two signals, B and pB, as shown in the

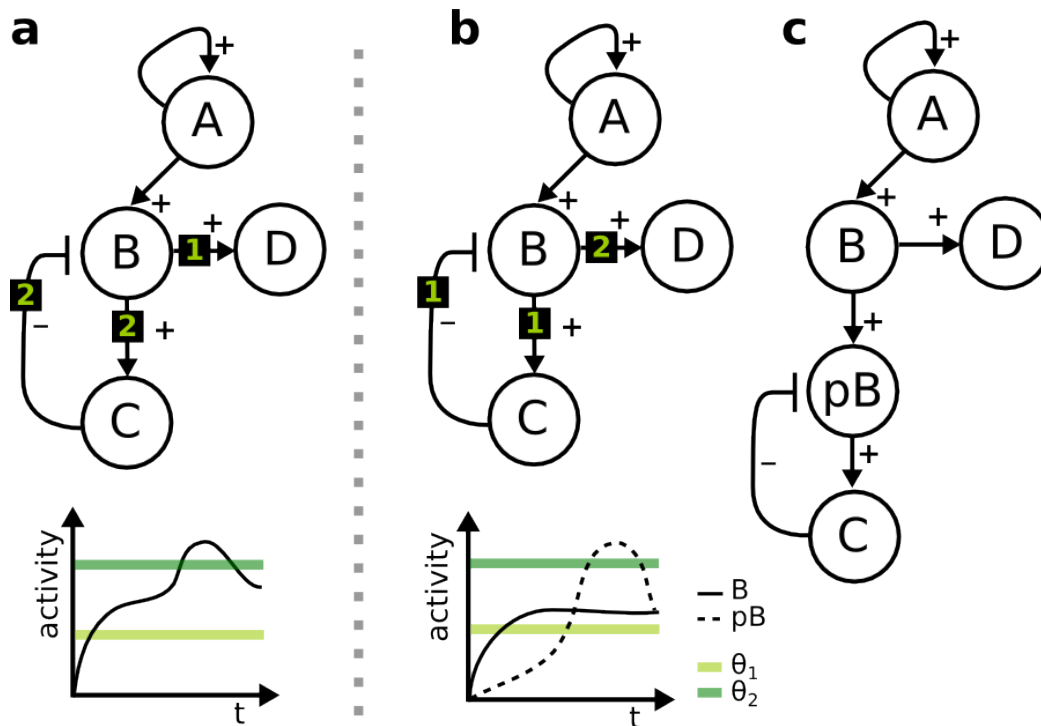


Fig. 3.3. Activity levels of a component and the thresholds of the outgoing edges affect the dynamical behavior. **a** Toy model with component B having three levels according to behavior in the diagram below. Thresholds θ_1 and θ_2 separate low, medium, and high activity of B. **b** Toy model with component B having three levels derived from the behavior in the second diagram. **c** Toy model with two separate components B and pB with two levels each, where θ_1 distinguishes low and high activity of B and θ_2 distinguishes low and high activity of pB.

diagram in Figure 3.3 **b**, the assignment of the thresholds to the components would need to be switched to capture the oscillations in Bp. The resulting interaction graph in **b** cannot reproduce the dynamics in the data, since the inhibition from C on B prevents B to pass the second threshold for activating D. In this case, the component B should be split into two components B and pB, shown in Figure 3.3 **c**.

In this step, for each component $v \in V$ the number of levels and for target u the respective threshold $\theta(u, v)$ is set. In case none of the component has more than two levels, the system is a Boolean network and all thresholds are 1.

3.1.3 Interactions and labels

Determining the interactions of the components within a network is a hard problem. This is due to the fact that the majority of the methods for obtaining data only record the activity of components, but cannot capture the interaction itself. For this reason, network inference is a very active field of research, where algorithms try to infer interactions from data based on correlation by using unsupervised or (semi-)

supervised learning approaches. Although high-throughput experiments deliver more and more data with increasing resolution, these methods still struggle as the number of interactions that can be inferred exceeds the number of independent measurements resulting in an underdetermined problem [24].

Here, I only look at models with a relatively small number of components, where it is possible to scan the literature for prior knowledge and data to construct the network. When trying to find an interaction between two components, one has to distinguish between direct or indirect effects. For example, in Figure 3.4 component A is activating both B and C, and it is known that B is directly activated by A. Whether or not A or B then activate C is uncertain, which can be expressed as edge labels on the connection. An alternative would be to simplify the model to A and C, in case the wiring between them is not of interest.

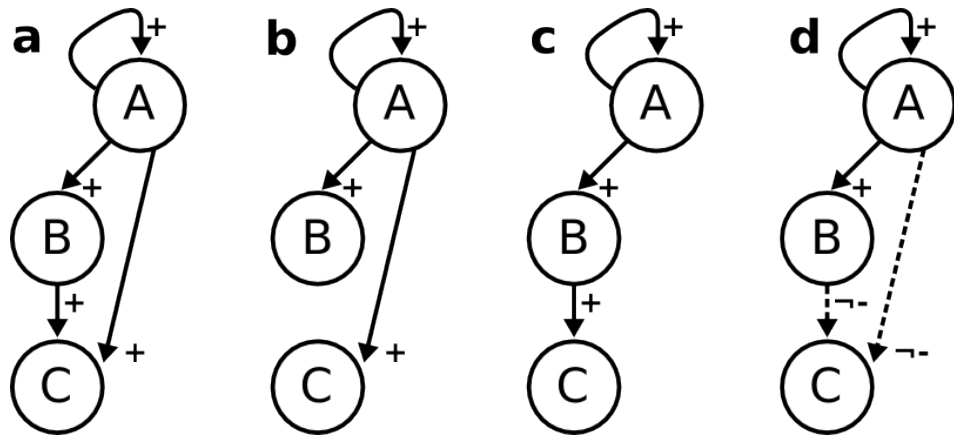


Fig. 3.4. Direct and indirect interactions of A and B on C. In **a** both A and B show direct interactions on C, in **b** only B directly activates C and in **c** only A acts on C. This uncertainty can be denoted in the graph **d** using edge labels.

The decision, whether a correlation in the data is a direct or indirect effect needs to be evaluated for every interaction. Experiments that show binding of components, such as pull-down experiments, are helpful to prove direct physical interaction, e.g. for scaffold proteins or complex formation. For other interactions such as modifications like phosphorylation it is often not possible to prove a direct effect. In that case, I either rely on a broad body of literature, simplify the model to a level where the indirect effect is a sufficient level of detail or mark this connection as uncertain using the edge labels introduced in Section 2.4.

Adding the interactions E and their labels l to the selected components V , I define the regulatory graph of the system $\mathcal{R} = (V, E, l)$.

3.1.4 Regulation constraints

In logical modeling, the regulation of each component by its predecessors is defined by logical equations. In case a component only has one predecessor, the logical equation either copies the predecessor's value for an activation, or is defined as its negation for an inhibition. However, if a component has more than one predecessor, the logical equation needs to be derived from biological knowledge.

There were methods developed to infer these equations automatically especially for large networks, where literature research would be too exhaustive or in case no biological information is present. An example is the activation-inhibition function by Martin et al. [67], where edges with the same sign are connected with the OR operator and edges with different signs are connected with the AND operator. For large datasets it is possible to use inference algorithms such as REVEAL, where unsupervised learning using support-vector machines fit the logical rules to the data [58]. The aim of these methods is to find one model that fits the data, but they do not account for uncertainty and prior knowledge. Also, they do not enumerate alternative models that fit the data equally well.

Here, I do not want to make assumptions but explore the uncertainty that is present in the data. Thus, I only create logical rules if they can be derived from biological information. A rule of thumb is given by Albert et al. [2]. It says that regulators that act independently are connected by the OR operator, whereas conditionally dependent regulators are connected by the AND operator. An example for dependent regulators would be a dual phosphorylation of two distinct kinases for full activation of a protein. An OR connection could be a binding site of a protein, for which two regulators compete. In general, the majority of entities in biological systems are strictly controlled by the interplay of activators and inhibitors, i.e. in signaling processes mainly by kinases and phosphatases [50]. Thus, a generalization to one rule that applies to all components can lead to biologically incorrect mechanisms and it is important to make use of available molecular biological information to derive the logical rule of regulation.

The result of this step is the definition of the logical functions f or constraints on parameters, which determines the parametrizations \mathcal{K} of the system.

3.2 Objective formalization and system adaptation

In this section, the biological objective of the study is incorporated into the modeling process. A model can be used for different aims, which influence the PKN built before, e.g. by adding further components or edges. For example, I present a workflow to integrate two models into one and then add crosstalk to the system, see Thobe et al. [98]. There are other interesting questions to be studied, like finding driver mutations or testing the influence of drugs on a network (Fig. 3.5). As shown in Figure 3.1, this step adapts the PKN and results in the generic model pool \mathcal{K} .

3.2.1 Investigating crosstalk between models

With rapidly growing technical progress and more accessibility of experimental data, more and more models are built to decipher the mechanisms that control cellular processes. However, in order to obtain a more global understanding, techniques for integrating validated models to more comprehensive systems are of interest [88].

Therefore, one objective in the toolbox is to couple two existing models and take uncertainties concerning the crosstalk into account, which was published in [98] and passages were adopted from the paper. The approach was developed to model coupling with different requirements in mind. First, the method should allow to decide which characteristics of the original models should be preserved in the integrated model. Second, the method should be able to handle uncertainty w.r.t. to the crosstalk connections between the original models. Lastly, the constraints posed by the original characteristics as well as experimental observations pertaining the integrated system should be exploited to obtain a clearer understanding of the crosstalk, possibly linking particular edges to specific functionalities of the integrated system. While here the case of coupling two models is presented, an extension for several models is straightforward.

Model Integration Given two networks $\mathcal{R}^1 = (V^1, E^1, l^1)$ and $\mathcal{R}^2 = (V^2, E^2, l^2)$ with their parametrizations K^1 and K^2 , that are assumed to be validated and analyzed w.r.t. some features of interest. For each model, a set $\mathcal{P}_{1,2}$ of properties is required to be preserved by an integrated model. Here, both structural properties, such as involvement of components in feedback circuits, and dynamical properties, such as attractor characteristics or input-output behavior, can be considered. In application, these properties should describe experimentally validated behavior or characteristics of the systems that should not be lost when combining the models.

Model integration is accomplished in two steps: first, the regulatory graphs are combined to one network by merging identical components and adding crosstalk edges, and in a second step the model pool comprising all possible parametrizations consistent with this network is generated.

The coupled network $\mathcal{R} = (V, E, l)$ is defined in the following way. The component set V is given by $V^1 \cup V^2$, where it is assumed that vertices that represent the same biological component coincide in both original models, i.e., they are merged within the integrated model. Dependencies and regulations within the single networks are kept and additionally new regulations between components of the uncoupled networks are introduced, so $E = E^1 \cup E^2 \cup E^{new}$ with $E^{new} \subseteq V \times V \setminus ((V^1 \times V^1) \cup (V^2 \times V^2))$ the so-called crosstalk. Lastly, denote l^{new} the labeling of the edges from E^{new} . The labeling of the integrated network is defined as:

$$l(u, v) = \begin{cases} \neg+ & \text{if } (u, v) \in E^1 \cap E^2 \text{ and } l^1(u, v) = \neg+ \wedge l^2(u, v) = -, \\ \neg- & \text{if } (u, v) \in E^1 \cap E^2 \text{ and } l^1(u, v) = \neg- \wedge l^2(u, v) = +, \\ l^1(u, v) & \text{if } (u, v) \in E^1 \setminus E^2, \\ l^2(u, v) & \text{if } (u, v) \in E^2 \setminus E^1, \\ l^{new}(u, v) & \text{otherwise.} \end{cases}$$

For edges that appeared in \mathcal{R}^1 and \mathcal{R}^2 a new label is created. In case the edge labels are the same, the new label is adapted from the single models. In case the edge labels differ, the less restrictive label is used as shown above. Here, we assume that the nature of the signs are the same in both models, meaning that the same component A and B cannot be connected by an activating edge in one model and by an inhibiting edge in the other model. This would raise a conflict and the modeling process should be reconsidered.

To generate the model pool, the parametrizations K_v of the single models are kept for those components v that are not influenced by new edges or new edge labels as a result of the model integration or crosstalk inclusion. For all other components all parametrizations in agreement with the edge constraints are considered.

3.2.2 Finding driver mutations

The abnormal behavior of cancer cells is caused by mutations in the cell. Due to extensive sequencing of various cancer types it is known that mutations accumulate in the development of tumors often resulting in hundreds of mutations in a single

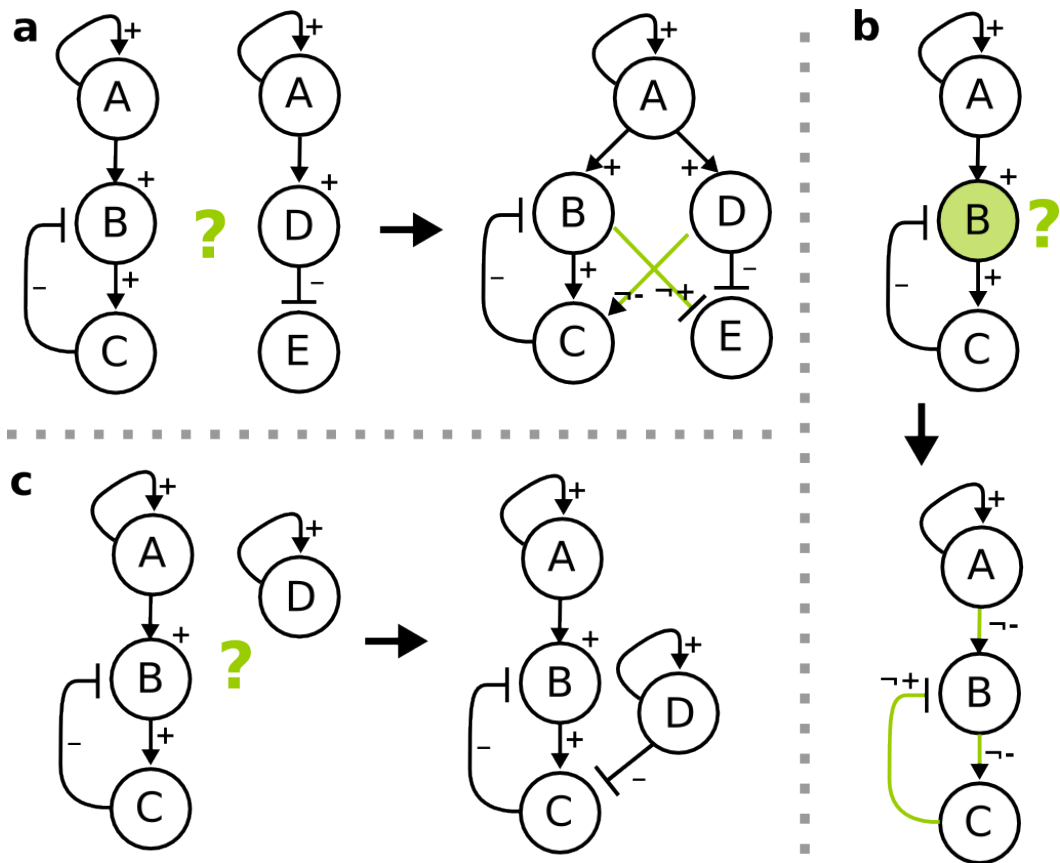


Fig. 3.5. Toy examples illustrate the different objectives of the toolbox. In **a** two pathways are connected with uncertain crosstalk, in **b** component B carries a mutation with uncertain effect on the upstream and downstream connections and in **c** the drug D is added to the network as an additional input of the system. Uncertain edges are marked green.

cell [13]. However, many of these mutations are not functional, so-called passenger mutations, whereas a few mutations trigger the disease, the so-called driver mutations [34]. These mutations cause over-activation and insensitivity to inhibitory regulators. Therefore, the proteins (called oncoproteins) are locked in an active state making them independent of its upstream regulations. Also, a loss-of-function mutation can occur in proteins that act as repressors of signaling processes, these protein are characterized as tumor-suppressors. Although many of these driver mutations have been described in general, it is often unclear which mutations govern a specific tumor cell [34].

In this approach, I account for mutations in components of a model with uncertain effect by setting all incoming and outgoing edges to *optional* even if these connections are textbook knowledge (e.g. see Fig. 3.5 **b**). Then, the network $\mathcal{R} = (V, E, l)$ is defined in the following way. The set of mutated components is given by $V^m \subseteq V$

with edges $E^m = \{(u, v) \in E \mid v \in V^m \vee u \in V^m\}$. The labeling l^m of edges in E^m is set to:

$$l^m(u, v) = \begin{cases} \neg+ & \text{if } (u, v) \in E^m \text{ and } l(u, v) = - \\ \neg- & \text{if } (u, v) \in E^m \text{ and } l(u, v) = + \end{cases}.$$

Thus, all incoming and outgoing edges are allowed to lose their function. In case an incoming edge is not observable in any model of the pool, the mutated component becomes independent from its inputs and the function of the component can either be set to 0 or 1 indicating a loss-of-function or constitutively active mutation, respectively. A change in the effects on the downstream targets of the mutated component can indicate a mutation in the protein structure and thereby affect its binding properties or a mutation in their active site pockets that leads to a dysfunctional protein.

However, I do not include the option for a gain-of-function by mutation, leading to new connections or a change in the sign of an edge. This is due to the fact that a gain-of-function would require to add optional edges from every component to the mutated component as well as an edge from the mutated component to every other node. This would lead to a dramatic increase in the size of the model pool. In order to keep the complexity as low as possible and considering that gain-of-function mutations are assumed to be rare, this option is excluded. An application for finding a driver mutation is illustrated in a case study in Chapter 5.

3.2.3 Testing the effect of drugs

With increasing knowledge and accessibility of detailed information on tumors, the treatment of cancer is changing. Besides standard treatment such as chemotherapy or radiation which do not account for tissue or tumor specific properties, drugs are designed to target the molecular causes of the disease [39]. This approach, called personalized medicine, is still in the early stages of development, since it requires detailed information about the regulatory processes in the cell and the effect of mutations on these [33]. In general, biological systems are structured to be robust towards perturbations, meaning that important events such as cell death or proliferation are controlled by multiple parallel pathways [51]. Tumors often acquire resistances towards the newly designed drugs by rewiring the signaling and thereby circumventing the blocked process [74]. Thus, combining two or more drugs has become a popular strategy to address the problem of resistances [103], but combinational therapy requires knowledge about the wiring of the network.

Here, the effects of drugs can be tested in combination on pools of models, without knowledge of the *true* network. For this approach, I introduce the drugs as additional input nodes to a PKN and connect them with an inhibitory edge to their target (e.g. see Fig. 3.5 c). For the network $\mathcal{R}' = (V', E', l')$ with f' as given logical equations, a set of components V is given by $V' \cup V^D$ with $v^D \in V^D$ is a set of drug components. The interactions of the network are given by $E = E' \cup E^D$ where new edges E^D are added, which contain the edge for self-activation to create the drug as input and an inhibitory edge from v^D to its target u , since the drug suppresses the function of its target. Similarly, the set of labels is composed of the labels of the original network and the additional labels for the drug components.

Usually drugs are especially designed to be a dominant influence on the target, meaning that it binds or modifies the target in such a way that it cannot interact with its former regulators or targets. In case the logical equation of a drug target is known, we can directly translate this dominant effect on the target u in a new logical equation:

$$f_u = f'_u \wedge \neg v_D.$$

If the drug is not dominant, which means that we are not sure about the mechanism with other regulators, the logical equation of the target is not defined and all possible regulations are generated in the pool. Moreover, it is possible that a drug affects off-target components in the system, then an edge from the drug to the potential off-target component is set to $\neg+$ and any regulatory context for that component is allowed.

In cancerous systems, one is usually interested in shifting the cell fate from survival to apoptosis, which can be a shift from an active steady state to an inactive for a specific set of inputs [35]. For our approach this means we want to assign the attractors of different input combinations to the model pool and investigate which models in the pool change to the desired attractor for which drug combination. In the analysis, the resulting subpools then represent groups of models that show the same asymptotic behavior for a certain drug combination, which could identify a biomarker for a drug combination. Also, the testing of drugs simulates perturbation experiments that can aid understanding the structure and finding properties that discriminate between subpools for experimental design. These experiments were shown to give valuable insights for complex signaling networks in cancer therapy [63].

3.3 Data formalization

In order to apply quantitative biological data to logical models, the data needs to be processed and encoded into temporal logics or other discrete formats. Originally, temporal logics were designed for electrical engineering, for man-made systems of known complexity and properties. Here, this method is applied to biological systems that carry a lot of uncertainty which is not straight forward and requires individual adjustments of the discretization, encoding and model checking process. Since the model checking process comes with high computational costs, all available information is incorporated to reduce the complexity of the problem.

Therefore in addition to the encoding of the data itself, the experimental setup and artifacts of the modeling formalism need to be considered. This information is encoded as a set of properties denoted by \mathcal{P} that are tested to satisfy models of the network $\mathcal{R} = (V, E, l)$ and formally create the refined model pool $\mathcal{K}(V, E, l, \mathcal{P})$ where $K \in \mathcal{K}(V, E, l, \mathcal{P}) \iff \forall P \in \mathcal{P} : TS(K) \models P$.

3.3.1 Incorporating genotype information

Often processes in cells with changes in the genotype are of interest, since they cause abnormal behavior that lead to diseases, such as cancer. Furthermore, due to a lack of information generic rather than cell line specific models are built. Under the assumption that specific models are derived from generic models by adapting node parameter values or edge labels, genotype information can be incorporated into the analysis for two scenarios.

Component mutations resulting in knock-outs or over-expression, can be modeled by requiring their value to remain constantly at 0 resp. 1 along all considered system trajectories. This can be phrased as additional constraints when encoding the experimental data used for filtering. In TomClass, the `Fixed` constraint is used to set a component to a certain value, whereas in Tremppi the user interface provides a setting called `experimental setup`. Consequently, this procedure ensures that the observed behavior is tested under the conditions imposed by the mutation.

Mutations can also alter the character of interactions, i.e., affecting the edges in the model. In case an optional edge is targeted by a mutation, we can find this as a result of the analysis provided that there is meaningful data. However, if a permanent edge is lost or a new edge is gained, the information must be directly included on the level of the edge constraints in the model definition.

3.3.2 Steady state assumption

Often, biological experiments are designed to measure the steady state of a system, which means that the system shows a stable behavior that is not expected to change without external influences. In general, this procedure is expected to yield more robust and reproducible results and is also prerequisite for other modeling formalisms or analysis methods, e.g. Modular response analysis [54] and Flux balance analysis [73]. Here, steady state data can be included into the analysis by interpreting it as a fixpoint of the model.

A fixpoint in logical modeling is an attractor of the system, which means that the system evolves to a single state which it cannot leave. In TomClass, fixpoints can be encoded in CTL formulas by using the delta constraint, see Section 2.7.1. For a fixpoint, the constraint $\Delta=0$ and the value of all components V need to be set. For example, to test whether there exists a fixpoint for the model $K = (V, E, l)$ with two components v_1 and v_2 being inactive, the formula $\mathbf{EF}(\Delta=0 \ \& \ v_1=0 \ \& \ v_2=0)$ is used. In case, we want every state or a set of initial states to evolve to this fixpoint, we additionally use the globally operator \mathbf{G} and apply it to all trajectories using \mathbf{A} . However, usually not all components of the system are measured, thus in TomClass, we can also test the stability of single components in the STG by only including a subset of components in the CTL formula.

In Tremppi, are two different levels to restrict the stability of a component. The first level is called `Ending` where we can only opt among `{stable, open, cyclic}` as described in the previous chapter. Thus, for testing a steady state, we select `stable`. In order to restrict the behavior of single components, the second level provides the options `{up, down, stay}` called delta constraints, which is assigned to every measurement and can be set to `stay` for stability of a component (see Sec. 2.7.2).

Applying steady state data as filter on a system requires caution, especially for systems with a negative feedback. First, steady state measurements are assumed to show the resting system, but there is no standard for how long a system needs to be untouched before it can be assumed to be in steady state. Moreover, an oscillatory structure usually does not lead to sustained oscillatory behavior in the biological system, but rather damped dynamics that can be interpreted as a fixpoint or look like a steady state in the measurements due to the resolution. However, negative feedbacks are a common motive especially in signaling processes and have an important function for terminating a signal or controlling the intensity.

In logical modeling, a negative feedback often leads to an oscillatory behavior. These cyclic attractors may conflict with fixpoint measurements and can lead to an under-representation of oscillatory systems in the model pool after filtering. Thus, the choice of testing a fixpoint on an oscillatory system must be taken with care. For example, in the time series data Experiment 1 (Fig. 2.3) of the toy example in Figure 2.2, the last measurement could represent a late time point where the system is assumed to have reached a steady state. In this case, it would be applied as a fixpoint constraint on the model pool. This procedure yields an empty pool, since none of the models is able to produce both an oscillatory transient behavior and a fixpoint.

3.3.3 Choice of strictness

When building a CTL formula from a data set, there are two levels where we decide how strict the formula is applied: the state formula and the satisfaction relation, Section 2.5. The satisfaction relations *ForSome* and *ForAll* are defined to determine whether only one initial state is required to be valid for the CTL formula or all initial states. Additionally, we can define whether an initial state fulfills the formula if one path is in agreement (exists operator **E**) or we require all paths to agree (forall operator **A**).

I refer these two levels as the *strictness* and their choice is not straight forward. This is due to the fact that biological information gained from experiments is incomplete, which means that not all components in the models are measured and not every transition is captured. Since we employ asynchronous updates to simulate the dynamical behavior of models, each state branches to all possible paths that result from the logical equations. In case we apply a CTL formula with high strictness using **A**(φ), it is only satisfied if every branch is valid for the CTL, although some branches might not be biologically realistic. This problem increases when components are not captured in the measurements and therefore are not restricted in the CTL formula.

In the toy example (Fig. 2.3), component A is not measured, thus applying the time series with high strictness would mean that for Experiment 1 the data would only be satisfied if from the states 1100 and 0100 every path satisfies the data. However, since A is the only activator of D, inactive A cannot produce these dynamics. In contrast, applying a CTL with a weak strictness, **E**(φ), means it is satisfied if only one branch from the initial state is valid for the CTL formula. This variant is more conservative, since only those models are rejected that fail to match the data with weakest constraints.

The choice of strictness for the initial states, *ForSome* and *ForAll*, is hard since we often only measure a subset of components. Since we generate every possible initial state in the STG, there might be a state which is not biologically feasible and could conflict with the CTL formula. However, most measurements are taken from cell populations and not single cells, showing the observed behavior for varying initial states. Thus, the high strictness can reflect a robust behavior of the biological system.

In practice, there is no rule to decide, which level of strictness is suited for a specific system and dataset. Often testing both options and exploit the resulting pool gives intuition about, whether a high or low strictness is more fitting. This testing also provides information about the system, in terms of how challenging it is to fit the data or other words how robust the system can produce certain dynamics.

3.3.4 Monotonicity of data

A further constraint that can be included for time series data is the assumption of monotonicity. This assumption implies that between two subsequent measured states the change is monotone, e.g. a component which is inactive in both states is not allowed to be active along the path between them. Without the monotonicity constraint there is no restriction on how and how often components change their value in the path that is satisfied.

In case there is a high resolution in time points of measurements, we can add the monotonicity constraint to a CTL to effectively reduce the pool to models with very specific dynamics. However, the disadvantage of enforcing monotonicity is that measurement errors cannot be compensated by the dynamics. Similar to the level of strictness, there is no strict rule for when to enforce monotonicity. Also testing monotone as well as non monotone versions of the CTL formulas can give information about the dynamical properties and robustness of the models.

3.3.5 Qualitative observations

Besides experimental data, general qualitative observations or information can be included as properties. Often basic behavior can be derived from the purpose of the modeled system, e.g. the quiescence state of the system. We argue that a signaling process is supposed to be inactive in the long term without any signals, since this is the general understanding of the biological function of signaling processes. In such kind of systems, we can include a so-called trivial fixpoint in our studies, where we

assume that the system should reach a fixpoint with inactive output if all inputs are inactive.

Another example for qualitative observations are oscillatory systems such as the cell cycle or the MAPK pathway. These system were exhaustively studied and proven to show a cyclic activation and inactivation of its output upon stable activation of the input [50]. Thus, even though we might not have explicit data at hand, we include such observations into our study as additional CTL formulas.

3.3.6 Transfer properties to a higher dimension

A special case is given for the objective to analyze the crosstalk between two existing models \mathcal{R}^1 , \mathcal{R}^2 by integrating them into one system \mathcal{R} . Here, the integrated model will have a higher dimensional state space than either single model. In order to interpret the properties \mathcal{P} within this new context, we may have to translate them into this new setting. This might not necessarily be straightforward.

For example, consider that an attractor A^1 of the model \mathcal{R}^1 should be preserved. One possibility would be to demand that A^1 is the projection of some attractor of the integrated model. A weaker condition would be that A^1 is an attractor of the state transition graph derived from projection from the state transition graph of \mathcal{R} . Also it is important to consider whether a property is an observed behavior for the elements of the single model in context of the joint system or might be an artifact of the isolated model. In application, the decision on how to translate the properties might be supported by biological knowledge or reasonable assumptions.

3.4 Model pool analysis

After filtering the generic pool for the data encoded as properties a reduced pool is determined, which can contain up to hundreds or thousands of models depending on the study. If the number of model in the so-called specific pool is very small, the analysis can be done by hand. However, often the number of models is too high to to be analyzed by hand, but require customized tools to extract interesting information. I evaluate the specific model pool by employing two different analysis tools, a statistical analysis and an exact analysis. Depending on the aim and the pool size, either or both analyses can give new insights into the biological processes.

3.4.1 Statistical analysis

The statistical analysis explores the difference between the generic pool and the specific pool, since this difference is strongly implied by the data. Here, both the reduced and the initial pools are evaluated statistically using Tremppi [91], as described in Section 2.7.2. Here, edges that are enriched or under-represented in the filtered pool in comparison to the initial pool are identified, as presented in a former study [98]. The visualization using Tremppi's regulations report provides an overview of the changes in the pool. General trends can be identified aiding the further analysis process. Especially large model pools can be compared easily as presented in the case study in Chapter 5.

Also, one can derive the most likely model from the analysis, which means that edges that are over-represented in frequency and impact are included and under-represented edges are discarded. Another result is to identify potential edges to be used for experimental design. For example, connections that have a medium frequency split the specific pool in two groups: those having the edges and those lacking the edge. Thus, testing the presence of this edge in experiments can reduce the pools size further.

3.4.2 Exact analysis

While the statistical analysis gives an overview about more and less frequent edges and their overall impact across the pool, it does not provide information about the topology and the mechanisms of single models. Often, not only the most likely model is interesting for us, also the minimal model as well as other features, for example the mutual appearance of edges in specific models.

A second analysis can evaluate the model pool by classifying the models according to features we select. Here this classification is called *exact analysis*, since the output give us information about the distinct models in the pool instead of a statistic across the pool. There are two options to implement this analysis. Either the model checking software TomClass presented in Section 2.7.1 is used or the database created by Tremppi can be analyzed using SQL queries directly. In case, one is only interested to do an exact analysis for a model pool, the workflow from model building to analysis can be implemented in TomClass. However, both analyses are of

interest, the model can be implemented into Tremppi and then the following SQL query to create the classification equivalent to TomClass:

```
SELECT features COUNT(*) FROM Parametrizations
WHERE datasets >0 GROUP BY features
```

where `SELECT features` defines the selected attributes we are interested in. Here, these attributes mean everything that is assigned to a model in the database `Parametrizations` created by Tremppi, i.e. \mathcal{K} , \mathcal{P} , the impact $imp_K(u, v)$ (see Section 2.7.2), the edge label l , and the indegree of each component. `COUNT` is used to determine the cardinality of each subset. The option `WHERE datasets >0` (which is equivalent to `cost!=0` in Tremppi) restricts the selection to models that are in agreement with the properties derived from data and `GROUP BY features` creates the classification. For the analysis I used the SQLite Manager¹.

The results of this analysis strongly depends on the features selected for the classification. For example, the model with the lowest number of optional edges to fulfill a certain set of properties can be determined, or edges identified that are necessary to be present in every model of the selection. Also models that cannot fulfill a certain property can be explored by setting `WHERE datasets =0`. Examples for the application will be given in the case studies.

¹<https://github.com/lazierthanhou/sqlite-manager>

Implementation in Tremppi

In this Chapter, the implementation of my workflow from Figure 3.1 in the software Tremppi is illustrated for the toy example used in Chapter 2 (Fig. 2.1). The software was developed by Adam Streck [91] and is available through the github platform. The user interface of the software is build in Javascript, thus it uses the browser.

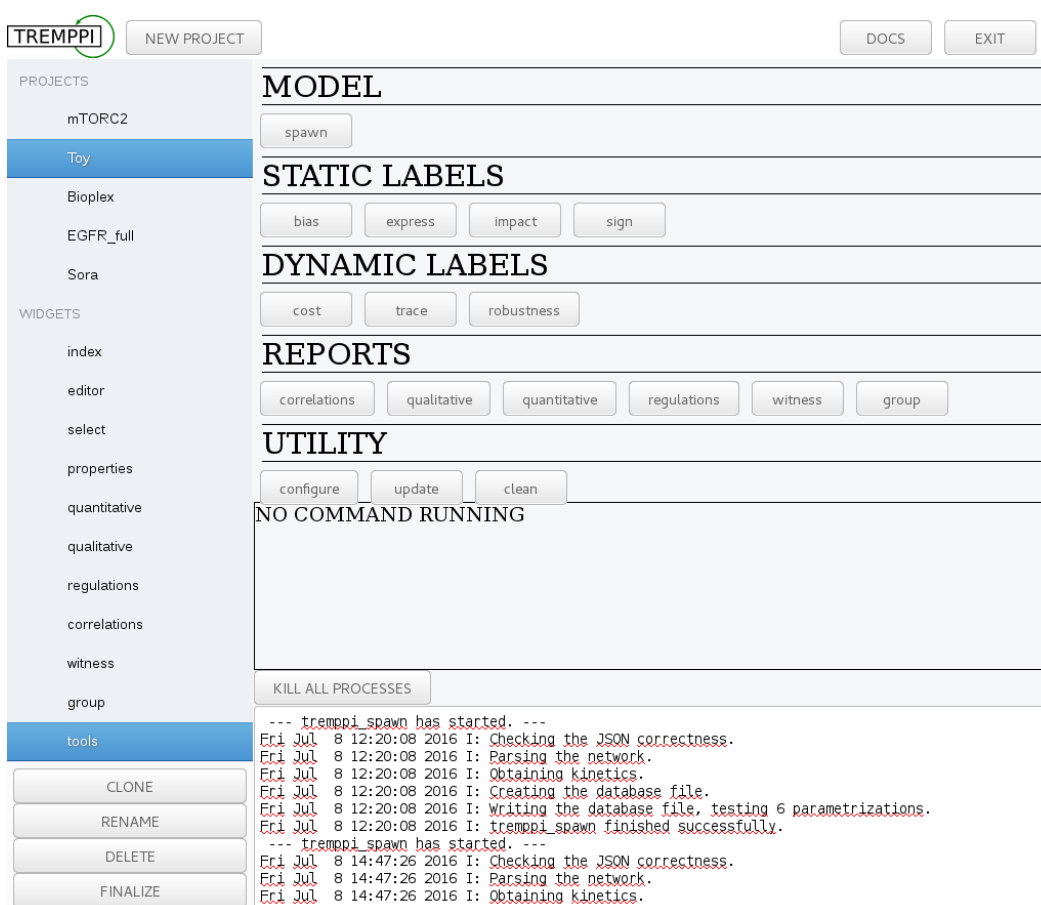


Fig. 4.1. Screenshot of Tremppi software showing the structure of the user interface with active tools section, where the calculations are being triggered.

Software description The interface has a menu on the left, where the first section PROJECTS lists all studies of the user and one can switch between them by simply clicking on the name, see Figure 4.1. Within each project, the menu section WIDGETS includes the options to build the model, enter data and observe results of different

analysis options as described in Section 2.7.2. Since we are not using all options in my studies, I will focus on those used in the case studies in the following chapters.

Before we present how the model is edited and analyzed, the main section for performing calculations is shown, the `tools` section. Here, operations concerning the whole project like cloning, renaming or deleting the project, updating the software and running commands on the models are triggered. The different commands are grouped in different categories as shown in Figure 4.1. The specific commands will be explained with their application in the following.

4.1 System initiation & objective formalization as PKN

In the presented pipeline, the system initiation and the objective formalization result in the prior knowledge network. In order to implement this network into Tremppi, the decisions on model boundaries, activity levels etc. as well as the question of interest must have been made before.

Then, the first step for implementing a study is to create a new project by clicking on the button in the upper left corner of the interface and name the project, e.g. in this case “Toy”. In the Index environment, a description of the project can be given.

Model building in editor For defining the logical model or the pool of models, all components $v \in V$ are added in the `editor` by first clicking on the +Add button and then clicking at the desired position in the grey window. Then the components can be renamed, a maximum activity value assigned and repositioned within the window. Additionally, the logical function can be added as `Constraint`, where we enter the known parametrizations of the component. For the toy example, components A and D only have one regulator each where the respective edge label is always observable. Therefore, the logical equation is already specified. Component B is supposed to have the logical equation $f_B = A \wedge \neg C$, thus we enter the parametrization:

$$A : 0, C : 0 = 0 \ \& \ A : 1, C : 0 = 1 \ \& \ A : 1, C : 1 = 0,$$

where the known regulatory context for B is defined. For example, the first expression means that $B = 0$ if A is 0 and C is 0. For component C the regulation is uncertain, hence we do not define the regulatory context.

In the next step, the edges $E \subseteq V \times V$ are drawn between the components by clicking on +Add then clicking on the source and subsequently on the target component in

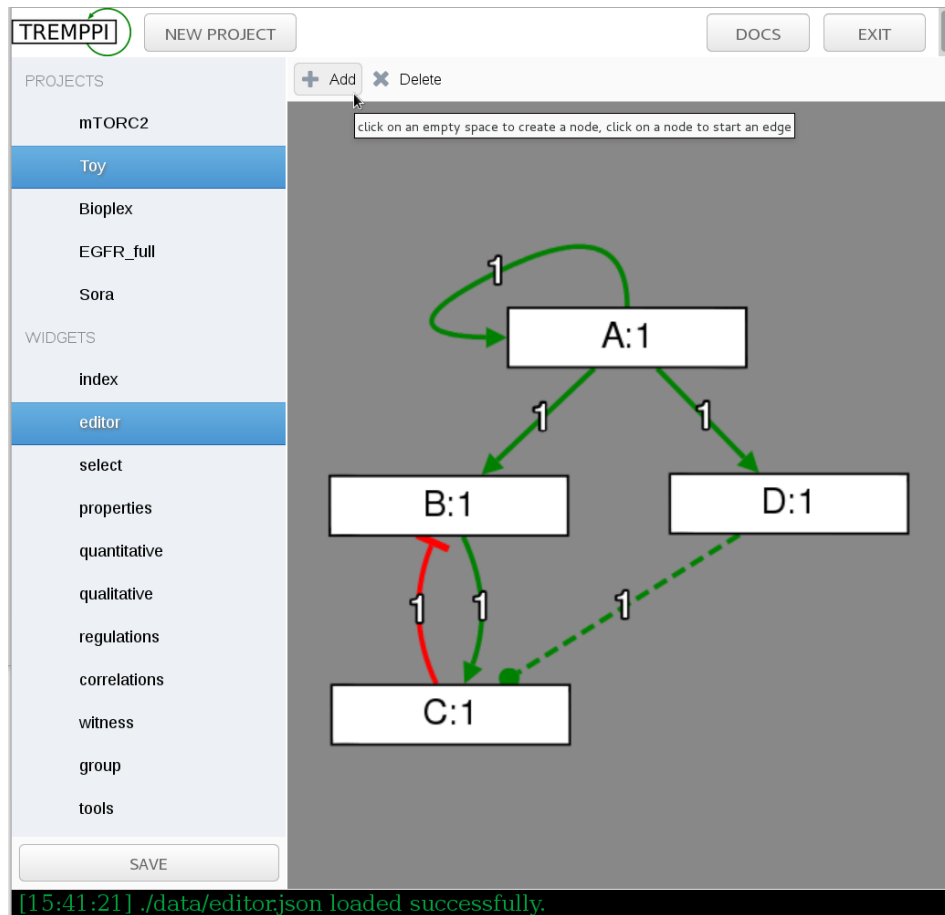


Fig. 4.2. The regulatory graph is defined in the editor by adding components and connecting them with edges.

the window (Fig. 4.3). To each edge a value is assigned, which represents the threshold for its activity. The default is 1, but the value can be changed by clicking on the respective edge. Figure 4.3 shows the option to change the threshold and additionally the edge label is selected in a drop-down menu. The default setting is Free, thus allows for any sign and presence. After defining the components, edges and edge labels, the graph needs to be saved by clicking on the button.

4.2 Data formalization for filtering the generic model pool

After entering the graph in the editor, the PKN is defined. For building the generic model pool, we need to go to the tools section and click on spawn. The command line located in the grey box in the center of the page shows the progress while the program is calculating all possible parametrizations based on the graph. The output of the program is presented in the white area at the bottom of the page, where details

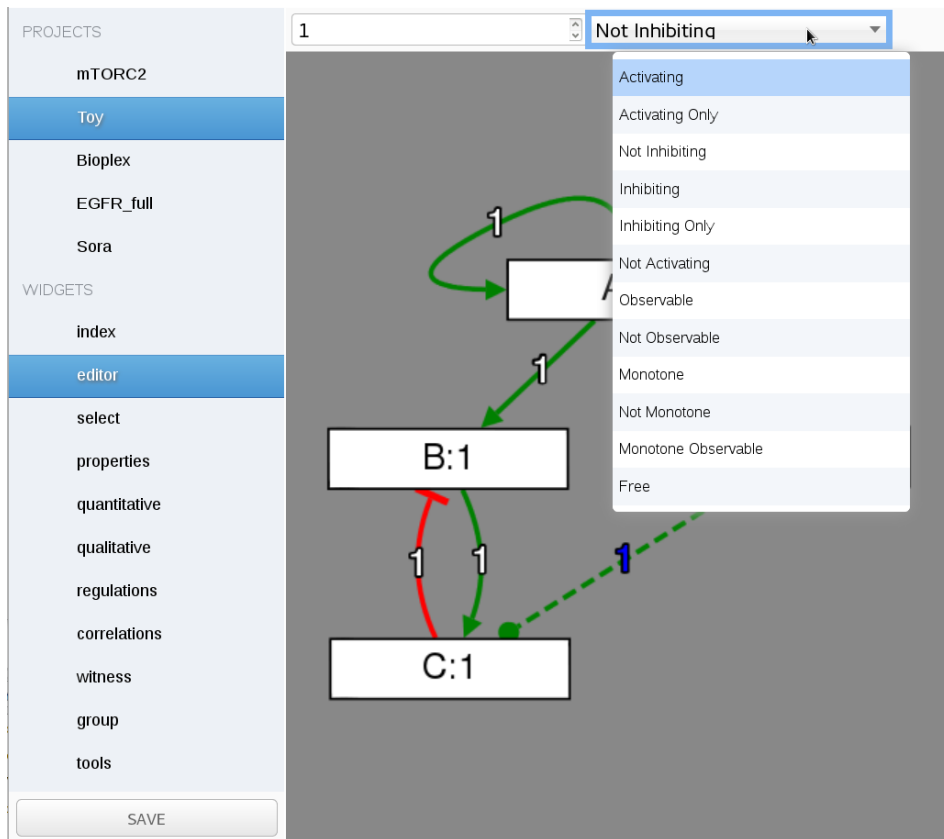


Fig. 4.3. Each edge need to be assigned with an edge label in the editor. The selected edge has a blue marked threshold value.

of the spawn function are reported. As shown in Figure 4.1, the tool enumerates the networks, generates the dynamics and stores the model in a database file. Also, in the same output is declared how many parametrizations were created, thus how many models are in the model pool. For our toy model, the generic model pool contains three parametrizations.

Data encoding in properties In the next step, we want to test the models for dynamics observed in experiments in order to find those models that are able to reproduce the biological behavior. For this purpose, the data needs to be preprocessed depending on the kind of experiment performed (see Sec. 2.5). After processing and discretizing the data, we implement each experiment as a property P in the `properties` section shown in Figure 4.4.

In Section 2.7.2, I explained that Tremppi translates a data set into a property by defining four elements: the sequence of measurements, the sequence of delta constraints, the ending and the experimental setup. In the user interface, the `properties` screen is split in two areas. In the upper area, a property is introduced by clicking on the +Add button or an existing property can be duplicated. Then

Fig. 4.4. In the properties section experimental data can be defined as a property. The upper table gives an overview on the different properties created with its global parameters. By clicking on a property, the details are shown in the lower table. Here, measurements and delta constraints can be assigned.

the global settings of this property like Name, Ending and experimental setup are defined. In the area below, the sequence of measurements for this property is defined as a table with two columns for each component to set the value and the delta constraint to the next measurement. A measurement is created by adding a new row to the table.

In Figure 4.4, the measurements from Figure 2.3 are used to illustrate the implementation in Tremppi. For each of the two experiments, every combination of ending and monotonicity constraints are tested. Thus, we get a non monotone and transient, non monotone and stable, non monotone and cyclic property and the same with monotonicity between the measurements. The monotonicity constraint is entered with every measurement as shown for property Exp1mon in Figure 4.4 where either up, down or stay can be selected.

For testing these properties on the model pool using model checking, the properties need to be marked by a checkmark and saved. Then, the model checking process is initialized in the tool section using cost, which assigns a cost value (described

in Sec. 2.7.2) to each property to every model. In case there exists a path in the transition system of a model that satisfies the property, a positive value is assigned in the database. Otherwise, the value 0 is assigned.

4.3 Analysis of the specific model pool

So far, we created the generic model pool and assigned experimental data as properties to each model. For creating and analyzing a specific model pool, we need to select which property or properties should be valid for the models in the specific pool. For example, we could investigate the same system in two different cell lines with multiple experiments each. Then we would create two specific model pools with different properties.

In Tremppi, the selection section is structured as a table, where each label annotated in the database has one column and the selections are added as rows. As described in Section 2.7.2, there are many different labels that can be assigned, whereas the parameter value for each component is annotated by default. After calculating the cost, the properties are added to the table as shown in Figure 4.5 for the toy example. Another default setting is the selection all shown in the first row of the table, which is the selection of all models in the pool, without a restriction on any label.

Name	Cost(property)						
	Exp1	Exp1mon	Exp1monSS	Exp1monOsci	Exp2	Exp2mon	Exp2monSS
<input checked="" type="checkbox"/> all							
<input checked="" type="checkbox"/> Exp1	10						
<input checked="" type="checkbox"/> Exp2					10		
<input type="checkbox"/> Exp2ss							
<input type="checkbox"/> Exp2mon						10	
<input type="checkbox"/> Exp2monSS							10
<input type="checkbox"/> Exp1ss							
<input type="checkbox"/> Exp1monOsci				10			
<input type="checkbox"/> Exp1osci							
<input type="checkbox"/> Exp1monSS			10				
<input type="checkbox"/> Exp1mon		10					
<input type="checkbox"/> Exp2osci							
<input type="checkbox"/> Exp2monOsci							

Fig. 4.5. In the selection section the specific model pools are defined by opting for a combination of properties.

For creating a new selection, the +Add button or a duplicate of an existing selection can be used. The new selection then can be named and labels can be restricted according to the rules presented in Section 2.7.2. Here, we only select for the satisfaction of a property by entering !0 in the respective field, meaning we select all models that agree with that property. In the example, we create a selection for each property, by marking those we want to analyze in the next step and finish by saving.

Statistical analysis in regulations The analysis of a selection is created in the tools section as reports. As described in Section 2.7.2, there are different kinds of reports available in Tremppi, here we focus on the regulations report. In order to create this report, the statistics of the selected model pool need to be calculated by running the static labels impact and sign in the tools section. After that the regulations report creates a file for each selection in the regulations section shown in Figure 4.6. In the example, we analyzed the selections all, Exp1 and Exp2.

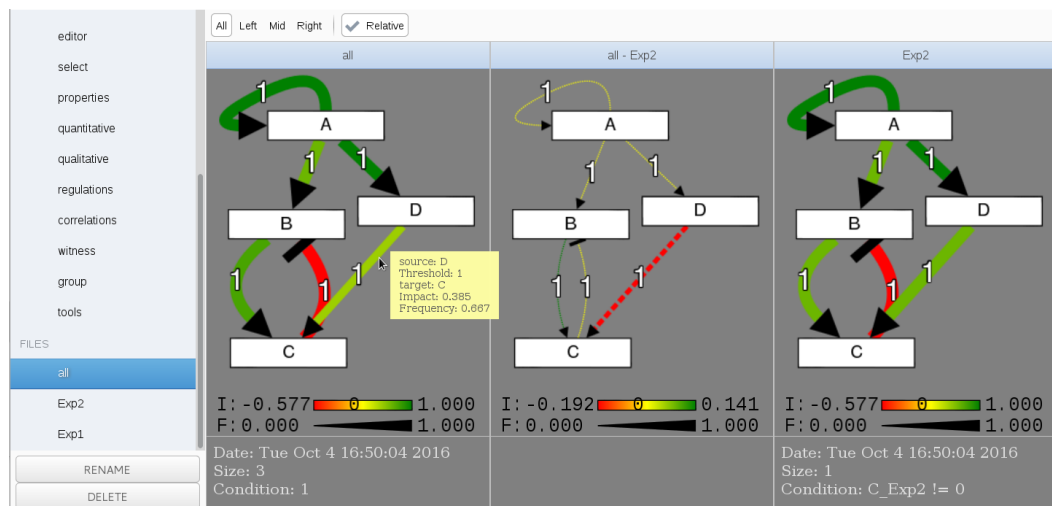


Fig. 4.6. Regulations report in Tremppi. On the left, the files for visualization can be selected, where two files can be compared with each other. The window gives a graph for both selected files and the difference between them.

The results are visualized as a graph when clicking on one of the files. The graph shows two different statistics of the pool: the impact and the frequency of an edge (formal definition in Sec. 2.7.2). The impact is presented as the color of an edge, ranging from red for inhibiting edges (maximum is -1) to green for activating edges (maximum is 1). The impact describes the correlation between the components connected by an edge, thus if the value is 1 it is fully correlated. This value is only possible if there is only one incoming edge on the target (e.g. in Fig. 4.6 component

D). The average frequency of an interaction in the pool is visualized by the thickness of the edge.

Above the graph, there is the option to show either All, Left, Mid, Right which defines how many windows are visible. The reason for these option is that one can either look at one selection only or compare two selections with each other by calculating the difference between them. This comparison is made by clicking on one file with the right mouse button and on the other file with the left mouse bottom. In the example, we chose the all selection for the left and the Exp2 for the right side. In case, we only want to look at either one of them, we could switch the view in the menu above the graph. The graph in the middle is the difference in both impact and frequency between the two selections. Here, the frequency can also be negative, which is indicated by dashed lines. The dotted lines in the difference graph show edges that are identical in frequency and impact in both model pools.

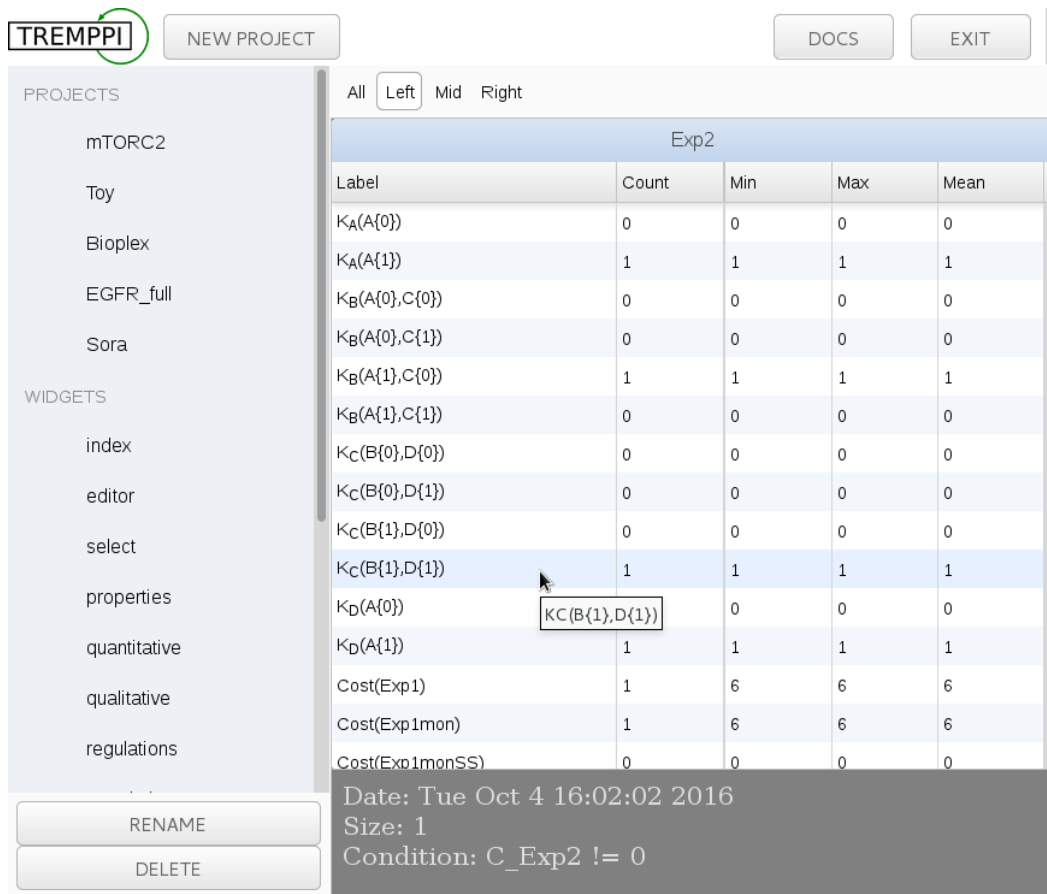


Fig. 4.7. Screenshot of Tremppi software showing the quantitative report for Exp2. In the table, the parameters for every component are listed and the resulting minimal, maximal and mean state of the component.

The text under the graph gives additional information about the date of analysis, the size of the specific model pool and the selection condition. For the specific

model pool Exp2 we find that the pool only consists of one model which contains the optional edge from D to C. Thus, this analysis is in agreement with the observation from Figure 2.4.

However, we cannot determine the logical equation from the statistical analysis, therefore the quantitative report is necessary. It is generated again in the `tools` selection and creates a file for Exp2 in the `quantitative` section. The result is shown in Figure 4.7, where the parametrizations of the models in the pool are listed. Since we only have one model left in the pool, it gives exactly the parametrization of this final model. Here, component C can only reach an active state if both B and D are active, listed in the row with label $K_C\{B\{1\}, D\{1\}\}$.

Investigating cell line specific EGFR signaling

The first application of the toolbox is published work that was done in cooperation with Adam Streck [92, 93]. In the paper, the focus was less on the workflow, but more on testing and comparing the performance of an improved version of Tremppi on a large data set [91]. The software and theory was done by Adam Streck shown in [92], whereas my work was the implementation of the model and interpretation of the results.

In this chapter, a broad background on the biological application we are regarding for all case studies is given. Then, in the second part an expanded version of the case study from [92] is presented.

5.1 Signaling in cancerous cells

Homeostasis is the ability of an organism to maintain stability in spite of changes in the environment through permanent self-regulation. Each cell is regulated by signaling pathways, which are the means to sense the environment, process the information and trigger the appropriate reaction of the cell. In order to turn a healthy cell into a cancer cell, Hanahan and Weinberg defined ten hallmarks describing necessary dysregulations of central signaling processes, such as resisting cell death, inducing angiogenesis or sustaining proliferative signaling [40, 41].

These dysregulations are caused by abnormal levels or formations of proteins resulting in loss-of-function or reduced activity for tumor suppressors or gain-of-function or overactivation for oncogenes. These abnormalities originate in changes in the proteinbiosynthesis of these proteins, which can occur in the genome due to mutations of the encoding gene [89], in the epigenome due to changes in modifications of DNA and histones [81] or in the transcriptome due to errors in the RNA regulation [20].

How these abnormalities orchestrate tumors is far from being understood, where the large variety in cancer types, the patient specific differences and heterogeneity

within one tumor is a major challenge in cancer treatment. One strategy is to identify dysregulated signaling processes and their drivers in order to use targeted treatment to trigger apoptosis in the tumor cells.

The so-called personalized medicine aims to predict the best treatment accounting for the genotype of a patient showing first successful applications, e.g. B-Raf mutation V600E [13]. However, not all oncogenes are druggable and often tumors are resistant or develop resistances during treatment. The reason lies in the robust and redundant wiring of signaling processes, thus targeting a single component can be circumvented [47]. In order to develop more effective therapeutic strategies, the wiring of the signaling and the effect of mutations need to be understood.

5.1.1 MAPK pathway

The mitogen-activated protein kinase (MAPK) cascade is an important pathway for cell survival, proliferation and resistance to drug therapy in cancer [23]. There are four independent MAPK pathways consisting of four different signaling mechanisms: the MAPK/Erk family or canonical pathway, and Big MAP kinase-1 (BMK-1), c-Jun N-terminal kinase (JNK), and p38 signaling families [9]. The canonical MAPK/Erk (extracellular signal regulated kinase) pathway integrates various internal stimuli, such as metabolic stress, DNA damage, and altered protein concentrations, as well as external signals like growth factors, hormones and chemokines [113].

Growth-factor signaling of MAPK/Erk The activation of the signaling cascade is triggered by tyrosine receptor kinases (RTK) such as the epidermal growth factor receptor (EGFR) upon binding of the ligand on the cell surface. The activation of the receptor causes an autophosphorylation of the intracellular domain of the receptor. Here, phosphorylated tyrosine-residues serve as docking sites for proteins containing Src homology 2 (SH2) or phosphotyrosine binding (PTB) domains, such as the adaptor protein growth factor receptor-bound protein 2 (GRB2) [82]. As visualized in Figure 5.1 a, the adaptor protein then recruits the guanine-nucleotide exchange factor son of sevenless (SOS), which then promotes a nucleotide exchange of GDP for GTP in Ras proteins [82].

GTP bound Ras has various downstream targets, one of them being the Raf protein family (A-Raf, B-Raf, c-Raf), which can be bound in a complex to further activate MAPK/Erk kinases (Mek1/2) by dual phosphorylation. Finally, these dual-specificity kinases recognize and activate the MAP kinases (Erk1/2) [23]. The activation of Erk regulates a wide range of downstream effectors involving both cytosolic proteins

and transcription factors to promote cycle progression and survival. Moreover, Erk terminates its own activation through negative feedback on Raf by causing the dissociation of GRB2 and SOS complex for Ras activation. Also, Erk regulates the activity of RTKs by inhibitory phosphorylation and diminished expression level of RTKs through transcriptional or post-transcriptional processes [94]. Thereby, Erk causes experimentally observed oscillations in the pathway [42].

MAPK in cancer The MAPK pathway is a very frequently dysregulated pathway in cancer cells, where it commonly acts pro-oncogenic but was found to function as a tumor suppressor as well [9]. Different members of the RTK family were shown to carry oncogenic mutations in cancer, e.g. EGFR in breast cancer. Ras was the first oncogene found in this pathway, but so far it was not possible to develop a drug for treatment. The more recent finding was an activating mutations in one of the Raf isoforms, B-Raf, which is very common in malignant melanoma (27%–70%) and also appears in colon, thyroid and lung tumors [31].

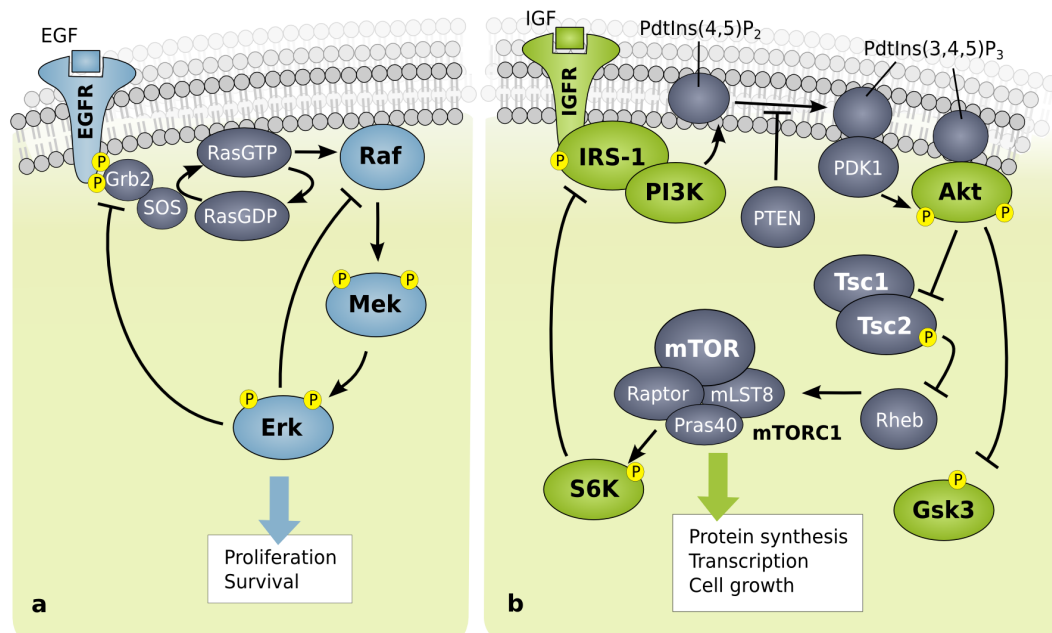


Fig. 5.1. Scheme of cellular signaling processes of **a** MAPK and **b** PI3K pathways.

5.1.2 PI3K pathway

The Phosphatidylinositol 3-kinase (PI3K) is a family of intracellular lipid kinases that regulate diverse cellular functions, such as cell growth, proliferation, differentiation, motility, survival and intracellular trafficking [27]. There are three classes (I-III) of PI3Ks, which are grouped according to their substrate preferences and distinct lipid products. In general, PI3Ks activate intracellular signaling processes by phos-

phorylating the 3'OH group of phosphatidylinositols, where each class is specific for different processes [19]. Class I PI3Ks are subdivided into two groups, according to their upstream regulation. Class I_A PI3Ks are activated by RTKs, whereas class I_B PI3Ks are activated by G-protein-coupled receptors [19]. So far, only class I_A PI3K signaling has been shown to be involved in cancers, therefore we focus on this specific class [82].

Class I_A PI3K signaling Class I_A PI3K is a heterodimer which is composed of a p110 catalytic unit and a p85 regulatory unit. Upon stimulation, some RTKs either activate PI3K directly, others like insulin and insulin-like growth factor receptors (IGFR) act through the adaptor protein insulin receptor substrate 1 (IRS-1) which then recruits and activates PI3K at the plasma membrane. Here, the p85 unit is crucial for binding to the phosphorylated RTK or IRS-1, which releases the inhibition of the p110 subunit [27].

PI3K then activates PdtIns(4,5)P₂ (PIP₂) to PdtIns(3,4,5)P₃ (PIP₃), which can be reversed by the phosphatase Pten. PIP₃ then recruits the multi-functional protein serine/threonine kinases of the Akt family (Akt1, Akt2, Akt3) and PDK1 to the plasma membrane [19]. Here, PDK1 phosphorylates Akt at T308, see Figure 5.1, and a second kinase mTORC2 phosphorylates Akt at S473, which will be the focus of Chapter 7. The phosphorylation at T308 is sufficient to activate Akt for its inhibition of the tuberous sclerosis complex 1/2 (Tsc) [75]. This process releases the suppression of the G protein Rheb, which then activates the mammalian target of rapamycin complex 1 (mTORC1) [64]. One main target of mTORC1 is p70S6K (S6K), which is able to phosphorylate IRS-1. Thereby the binding of IRS-1 to PI3K is disrupted and PI3K becomes inactive, thus creating a negative feedback [44].

PI3K in cancer There are estimates suggesting that mutations in one of the PI3K pathway components account for up to 30% of all human cancers [82]. Here, mutations and amplifications in one of p110 genes PIK3CA is very prominent oncogene, whereas a mutation of the p85 subunit is rare. Also PI3K's direct antagonist Pten is a well known tumor suppressor, which appears to be mutated in many tumors [82]. Further downstream in the pathway, Akt often is amplified in cancer cells. Since Akt controls cell survival, cell cycle, cell growth and metabolism through a number of substrates, it is thought of as a key player in tumorigenesis [27]. Along with Akt, mTORC1 has become focus in drug development for its role in inhibiting autophagy, increasing angiogenesis, and promoting the transcription of oncogenes [116].

5.2 EGFR signaling pathway study

This section presents a case study published with Adam Streck and Heike Siebert as conference and journal papers [92, 93]. In these papers, Tremppi and its implemented methods to effectively study biological systems were demonstrated. In the following, my contribution to these papers is presented where passages were adopted from the original paper and additional work is shown.

5.2.1 Motivation

In this section, we aimed at finding the cell lines specific wiring of signaling processes of the epidermal growth factor receptor (EGFR) motivated by a study by Klinger et al. [54]. This aim represents the basic objective of investigating an uncertain topology and mechanisms using the toolbox from Chapter 3, where the system initialization, data formalization and pool analysis are applied.

As described before, the receptor drives cell proliferation and cell growth, but is also involved in the regulation of cell death and is found to carry prominent mutations in cancer cells (B-Raf, PIK3CA). However, the exact topology of this regulatory system is not completely clear, not least since mutations can cause major changes in the inner regulations. Klinger et al. presented a combined experimental and theoretical approach to identify the cell line specific topology of the network starting from a literature model aggregating information from various sources. To this end, human colorectal cancer cell lines were treated with stimuli and inhibitors to produce a rich data set, where the experimental setup is outlined in Figure 5.2.

Experimental set up of perturbation experiments In detail, Klinger et al. used six cancer cell lines LIM1215, SW304, SW480, HCT116, HT29, and RKO to perform perturbation experiments with two stimuli, TGF α and IGF1, and four inhibitors: the Mek inhibitor AZD6244, the PI3K inhibitor LY294002, the GSK3 inhibitor SB216763, and the IKK inhibitor BMS345541. Each sample was pre-incubated with one inhibitor or DMSO for 60 minutes and then stimulated with one of the stimuli or BSA (Bovine serum albumin). The phosphorylation levels of multiple kinases in the EGFR pathway were measured 30 minutes after stimulation [54].

Within their study, several measurements were excluded from their analysis. The IKK inhibitor showed unspecific effects on Erk, also IKK and I κ B α were excluded. Moreover, the cell line RKO showed a very different behavior than all other cell lines,

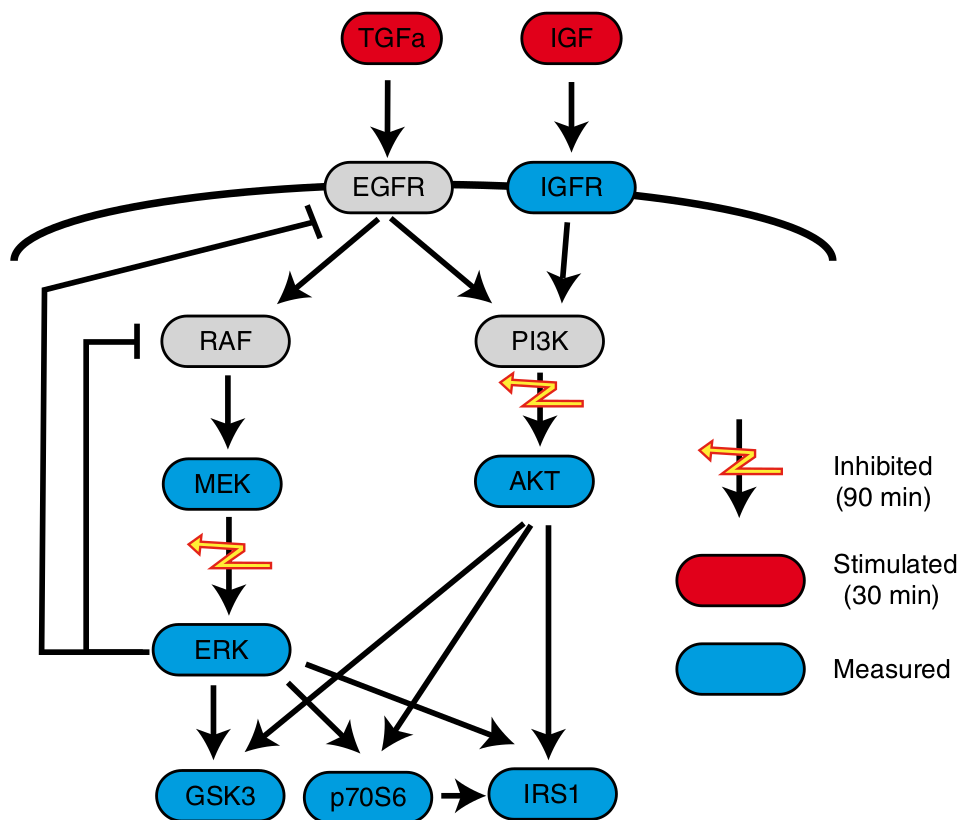


Fig. 5.2. Scheme of biological system and experimental setup in the study from Klinger et al. (figure adapted from [54]).

thus it was excluded [54]. For our model, we further reduced the data set by not considering all samples using the GSK3 inhibitors, since there is no downstream target measured to have a read-out for its effect.

Comparison of two modeling approaches and exploration of analysis options of our method

In Klinger et al., the EGFR signaling system was investigated using a semi-quantitative modeling approach, called modular response analysis (MRA). This approach was developed by Bruggeman et al. to calculate the response of a linear approximation of an ordinary differential equation model to a perturbation [8]. Klinger et al. created an algorithm that is able to include multiple perturbations and unobserved nodes. This algorithm starts with a literature-based network similar to our PKN and eliminates non-identifiable nodes. Next, the parameters of the system are calculated using MRA-based maximum likelihood to determine the weights of the edges. Subsequently, edges with a weight below a certain threshold are deleted [54]. The result is a reduced weighted graph for each data set they tested. The limitations of this approach are that it necessitates a steady state assumption for the data points, which can be seen as problematic due to the interplay of various

feedback effects [77]. In addition, while quite comprehensive, the method still relies on parameter estimation steps and statistical cut-offs.

We used the comprehensive data set provided by Klinger et al. in [54] to elucidate the underlying network structure of the pathway. For this aim, we generated and analyzed comprehensive model pools for the different cell lines and compared the results to the original publication. Thus, to maintain a maximum level of comparability, a steady state assumption for the data points was added. However, going beyond the study by Klinger et al., we additionally investigated the results for relaxed steady state or transient temporal constraints. A comparative analysis of the model pools was done, where the differences between cell lines was evaluated in not only network topology but also regulatory mechanisms generated by different genotypes.

5.2.2 Model building and data formalization

System initialization Based on the model of [54] we constructed a BN, depicted in Figure 5.3. We kept the original components and regulations, with a few exceptions. As the IGF1 stimulus is the only regulator of IGFIR we know that IGFIR copies its value and therefore we modeled the stimulation directly on IGFIR, removing IGF1 completely. Additionally, p70S6K is depicted as activator of IRS-1, however based on [95] we modeled it as an inhibition. The same for Akt which is known to repress IRS-1 indirectly through mTORC1 [95]. Note that these changes are to regulations of IRS-1 only, which is an output component and therefore can not affect the upstream feedback loops. Any resulting inconsistencies with [54] should therefore be localized to IRS-1. Since the data originates from cancer cells, we accounted for possible disruptions in the network due to mutations by not requiring regulations to be functional, i.e. activations are labeled as $\neg-$ and inhibitions as $\neg+$. However, stimuli and inhibitions as well as components with a single regulator (Mek, Akt) were set as always functional.

Objective formalization In the data there are two stimuli, TGF α and IGF1, and two effective inhibitors, Mek inhibitor AZD6244 and the PI3K inhibitor LY294002. There are two more inhibitors in the original data set on GSK3 and IKK, which were found to be non-effective and therefore neglected here. In our model, we define the inputs as $f_{TGF\alpha} = TGF\alpha$, which is set to 1 if the cell is stimulated with TGF α and 0 otherwise. IGFIR is set to 1 if the cell is treated with IGF and 0 otherwise. The inhibitors do not remove the targets from the system, only prohibit their effect on the downstream components. We therefore added them as additional input components

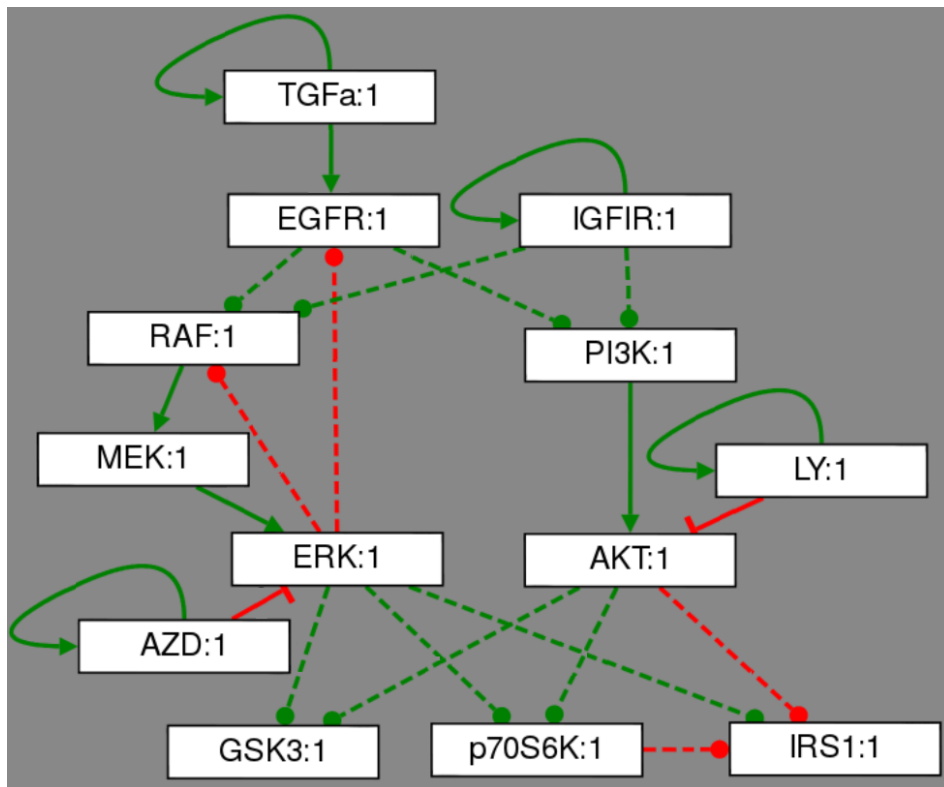


Fig. 5.3. Regulatory graph of the EGFR model in Tremppi. Solid lines mean that these edges are essential, whereas dashed lines show optional edges. Green and red color show activating and inhibiting effect, respectively.

LY and AZD, and modeled them analogously to stimuli. Additionally we set the regulatory functions $f_{Erk} = Mek \wedge \neg AZD$ and $f_{Akt} = PI3K \wedge \neg LY$ to enforce the correct inhibition semantics.

Data formalization In their experiments, Klinger et al. used a high-throughput immunoblotting method, called Luminex assay, which measures intensities of labeled antibodies that bind the phosphorylated components, showing their activity (for a detailed description see [54]). Here, we used a reduced data set containing experiments on five human colorectal cancer cell lines. Each of the cell lines was treated with each pairwise combination of one stimulus (TGFa, IGF, no stimulus) and one inhibitor (AZD, LY, no inhibitor), which were then compared to the measurements before treatment. Here, the configuration without any stimulus and inhibitor is not expected to change and is used as control sample. In the study of Klinger et al., the data is presented as a heat map for fold changes, comparing each measurement to the control sample (see Fig. 5.4).

Prior to their usage, the data needed to be processed to fit the Boolean formalism. The original data was kindly supplied by the authors, thus we were able to process

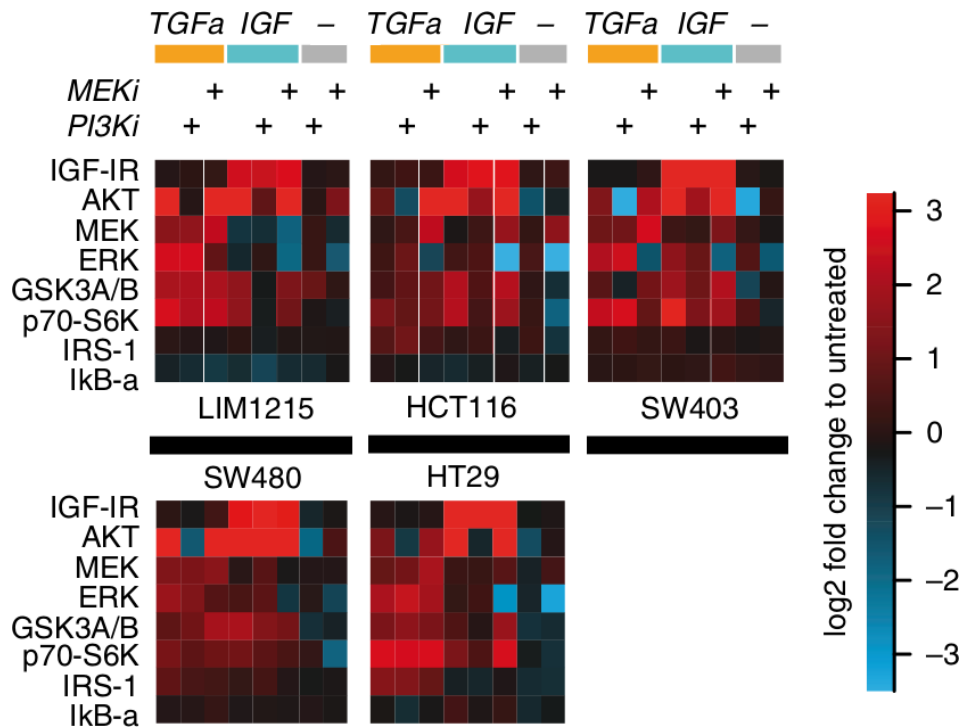


Fig. 5.4. Reduced heat map from Klinger et al. showing the fold change data of the perturbation experiments for the different cell lines (figure adapted from [54]).

the data ourselves. For some experiments multiple measurements were available, where the arithmetical mean was used. For the discretization we used ternary levels (see Sec. 2.5), since some of the values almost do not change between measurements in the data set being e.g. at a plateau and therefore should not be assigned with different states. To avoid this separation and in order to maintain a level of comparability with the original study, we focused on the fold change as a measure for discretization.

Here, we rely on an assumption that a fold change of value two or more is significant, which is to the best of our knowledge a common practice and in our case seems to produce a good separation. Since the focus of this study was to evaluate the effect of regulatory influences, we assigned Boolean values to the component measurements consistent with the nature of the fold changes found in the data. If we observed an increase by a factor of at least two, we assigned the value 0 to the measurement before and 1 after the treatment. Analogously, we encoded a decrease by a factor of at least two. If the change factor is less than two, we did not specify the value, but required the component to be stable, meaning that it either is constantly 0 or 1 throughout the measurement (for more details see [92]).

In this approach, the interpretation of qualitative dynamics heavily focuses on the component changes indicating actively regulated behavior, as was our intention.

Note that this approach therefore differs from the often employed interpretation of the Boolean component values as an abstraction for ranges of quantitative values. In our approach, the same quantitative value might be assigned different Boolean counterparts depending on the observed component behavior in the respective experiments. In our opinion, this does not pose a problem, since we are focusing on the qualitative dynamics and thus the values 0 and 1 can be viewed as labels of qualitative change, rather than ranges of quantitative values. Presumably, if a component can undergo both a significant increase or decrease in its activity, such mechanics should be allowed by the network without contradicting the effects of the regulations.

Tab. 5.1. Data set for cell line LIM1215 for the reduced set of treatments.

LIM1215								
Stimulator	Inhibitor	Type	Akt	Erk	GSK3	IRS-1	Mek	p70-S6K
0,1% BSA	DMSO	c	143	380	206	86	992	351
IGF 1	DMSO	t	7989	312	747	105	671	1277
TGFa	DMSO	t	2793	2806	983	104	3416	2708
0,1% BSA	AZD 6244	t	398	152	262	95	789	323
IGF 1	AZD 6244	t	6313	121	593	95	347	874
0,1% BSA	LY294002	t	168	518	471	94	1362	373
TGFa	LY294002	t	163	2863	1048	96	3596	1829
0,1% BSA	DMSO	c	0	-1	0	-1	-1	0
IGF 1	DMSO	t	1	-1	1	-1	-1	1
0,1% BSA	DMSO	c	0	0	0	-1	0	0
TGFa	DMSO	t	1	1	1	-1	1	1
0,1% BSA	DMSO	c	0	1	-1	-1	-1	-1
0,1% BSA	AZD 6244	t	1	0	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	1	0	-1	1	0
IGF 1	AZD 6244	t	1	0	1	-1	0	1
0,1% BSA	DMSO	c	-1	-1	0	-1	-1	-1
0,1% BSA	LY294002	t	-1	-1	1	-1	-1	-1
0,1% BSA	DMSO	c	-1	0	0	-1	0	0
TGFa	LY294002	t	-1	1	1	-1	1	1

The preprocessed absolute data is shown at the top and the discretized data in the bottom table. Note that the control sample is identical and repeated for the discretized data, since each treated and control sample are interpreted as a pair.

An example of the discretization is illustrated in Table 5.1 for the cell line LIM1215. The upper part of the table, shows the preprocessed data in absolute values, whereas in the lower part, the rows show the control (BSA, DMSO) and a treated sample in turns as discretized values. Since we always compare each treated sample to the control, the discretize values of the control sample differ. Note, that there are entries with -1, which encodes no change in this table and are later translated as

delta constraint stay. For this reason, -1 entries always come as a control-treatment pair. The full data sets are listed in the Appendix, see Figures A.1 to A.5.

5.2.3 Filtering and analyzing the cell line specific pools

After having resolved all the edge constraints, we obtained a model pool \mathcal{K}^l with 259,200 models using Tremppi. Note that as the inhibitors and stimuli are fixed components, they do not contribute to the size of the state space, which then only has $2^9 = 512$ states instead of 2^{13} . In the following steps, we filter this generic model pool for each data set separately in order to find cell line specific model pools.

Tab. 5.2. Results of applying data sets to the generic model pool.

Cell line	a Experiments				b Pool size		
	TGFa	IGFIR	AZD	LY	<i>transient</i>	<i>partially stable</i>	<i>stable</i>
LIM1215	1	0	1	0	180000	6100	40
LIM1215	0	1	0	1			
HCT116	0	1	0	1	129600	5580	2
SW403	0	1	0	1	180000	111000	840
SW480	0	1	0	1	136800	74670	36
HT29	1	0	1	0	163800	101010	216

a Some experimental set ups cause logical inconsistencies after discretization. **b** Sizes of a parametrization sets that match the data from all the consistent experiments for each cell. Monotone property sets are not listed as monotonicity did not cause any further reduction.

Discretization of data with different temporal constraints As we considered eight treatments for five cell lines, we obtained altogether 40 measurement pairs. In [54] the authors argue that at the time of the measurements the system is expected to reach a stable plateau. However, Figure S1 in [54] shows that the kinetics of some components have an unstable behavior after the time point of measurement. To investigate the impact of the steady state assumption, we created a *stable* and *transient* (i.e. not required to be stable) version of each time series.

Additionally, we were interested in effects of monotonicity constraints on the results. We therefore also considered for each property a version where all the components that are measured and not stable are required to be monotonous in their behavior. By combining the treatments, cell lines, and constraints we obtained 160 properties¹. However, we found that adding a monotonicity does not affect the restrictive effect of a measurement pair on the resulting model pool. Therefore, we focused on the stability constraint.

¹http://dibimath.github.io/CMSB_2015/properties.html

Initially, we found that each of the cell lines shows inconsistencies in at least one measurement pair, listed in Table 5.2 a. In each of these, the experimental set up requires a component whose activator was inhibited to undergo an activation, which is logically inconsistent. For example, cell line SW403 shows with IGF1 stimulus an over 4-fold increase in activity of Akt under inhibition of PI3K, its only activator. This is still comparably lower than the about 12-fold increase without the inhibition showing that the inhibitor is working, but the dose is not sufficient to lower the activity of Akt to the threshold of being inactive after discretizing. Since dose-dependent processes are not considered in this formalism, we removed the respective experiments from the testing set. After the removal, the number of sets of measurement pairs for each cell line reduced to seven except for LIM1215 where there were only six. Therefore, only 34 measurement pairs were used in the further analysis, yielding 136 properties when combined with the different temporal constraints. In Table 5.2 column *transient* shows how many members of \mathcal{K}^l fit all the measurements for each respective time series, which represents the weakest assumption concerning the stability of the system. Note that each set remains more than one half in size compared to the set of models consistent with the constraints derived from the network structure, suggesting that the topology itself already strongly determines the dynamics.

Tab. 5.3. Comparison of occurrence of different regulatory functions between the stable pools.

Target	Must	Klinger et al.	May	Match
HT29				
EGFR	TGFa	TGFa, Erk	TGFa, Erk	yes
Raf	\emptyset	EGFR, IGFIR	EGFR, IGFIR	yes
PI3K	\emptyset	EGFR, IGFIR	EGFR, IGFIR	yes
GSK3	\emptyset	Akt	Erk, Akt	yes
p70S6K	\emptyset	Erk, Akt	Erk, Akt	yes
IRS-1	Erk	Erk	Erk, Akt, p70S6K	yes
SW480				
EGFR	TGFa	TGFa, Erk	TGFa	no
Raf	\emptyset	EGFR, IGFIR, Erk	EGFR, IGFIR	no
PI3K	EGFR, IGFIR	EGFR, IGFIR	EGFR, IGFIR	total
GSK3	\emptyset	Akt	Erk, Akt	yes
p70S6K	Erk, Akt	Erk, Akt	Erk, Akt	total
IRS-1	Erk	p70S6K	Erk, Akt, p70S6K	no

The Must column contains the set of edges that are functional in all parametrizations fitting the data. The Klinger et al. column contains the ones reported in [54]. The May column contains the edges that are functional in at least one parametrization fitting the data. If May includes edges of [54], Match is set to yes. If Must and May are identical and match Klinger et al., Match is set to total.

Comparison of cell line specific model pools to results of Klinger et al. In [54] the modular response analysis (MRA) method was used to identify non-functional connections in the network for the different cell lines. Here, we aimed to compare the topologies of their resulting networks with the topologies that occur in our model pools. To improve comparability, we used a stability requirement for the measurements in each cell line to account for the steady state assumption in the MRA approach. The sizes of the parametrization sets are listed in Table 5.2-*stable*. Note that there is a much stronger reduction than in the transient case, suggesting that the stability requirement is indeed very strong for this network, presumably due to the negative feedback mediated by Erk. However, each of the resulting parametrization sets is non-empty, therefore we can compare which edges are required/allowed to be functional.

The results for two examples, SW480 and HT29, are shown in Table 5.3, where for HT29 the functions in the pool fit well to the results of Klinger et al. This means that in the pool for HT29 for every component there is a model which matches the regulation of the original paper. Although this seems to imply that there could be one model which matches Klinger et al. in all components, this was not true. For all other cell lines such as SW480 our results match [54] only in part. This is likely to be caused by negative feedback from Erk which is a source of instability in the Boolean framework, but in the real system may lead to damped oscillation and consequently to a quasi-stability. Additionally, the effect of Erk on IRS-1 creates an incoherent feed-forward motif, which was not captured in [54] as there the semantics of the regulations of IRS-1 are different.

Tab. 5.4. Presence of regulators in the individual cell lines.

Target	a: LIM1215-HCT116	b: HCT116-SW480	c: SW403-HT29
EGFR	differences in frequency	almost the same	no difference
Raf	LIM allows for 15 (out of 20) functions, HCT only for $y = 1$	HCT allows only for $y = 1$, SW for 15 functions	no difference
PI3K	strong increase in $y = 1$	differences in frequency	no difference
GSK3	strong increase in $y = 1$	no difference	almost the same
p70S6K	$y = 1$ appears	almost the same	almost the same
IRS-1	no difference	almost the same	almost the same

For each pair the difference of the first member when compared with the second member is described. The notation $y = 1$ is a shorthand for $K_v(s) = 1$ for any $s \in S$ where $v \in V$ is the target. For most of the cases, the same set of functions was present, but the frequency of their occurrence in the set differed.

Analysis of partially stable model pools As our method allows for testing transient states, and the time series measurement in Figure S1 of [54] illustrates that Akt and

Erk may not be in steady state at the time point of measurement, we also created a *partially stable* selection. Here, only experiments without stimulation were assumed to be in steady state, since their last treatment with an inhibitor was 90 minutes before measurement. Stimulated samples were allowed to be in a transient state since they were measured only 30 minutes after treatment. In our opinion, this scenario is a good compromise between being more biologically realistic than the *stable* constraint, but still observing a much stronger model pool reduction than the *transient* constraint (see Table 5.2), in order to be able to observe statistical effects. Thus, we used the *partially stable* constraint as the basis for the subsequent analysis.

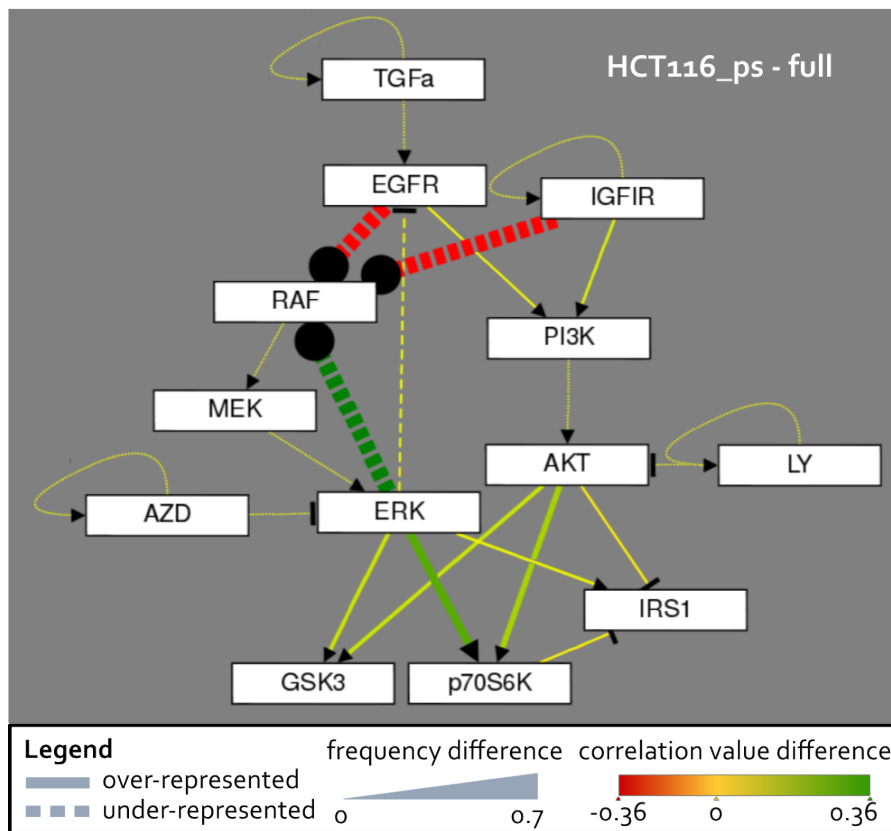


Fig. 5.5. Visualization of the statistical analysis of the model pool of HCT116_ps compared to the initial model pool. The incoming edges of Raf are strongly under-represented and their correlation for activating edges is negative, the negative edge from Erk shows a positive correlation.

Focusing on a comparison of the cell lines carrying different genotypes we expected to find topological and functional differences between the pools causing the observed variations in the measurements. The pools of all five cell lines were compared, resulting in ten different tables. We selected three comparisons in Table 5.4 to exemplary show the most deviating pair in **a**, the most similar pair in **c**, and an intermediate pair in **b**. Table 5.4 **a** shows the comparison of the pools corresponding

to LIM1215 and HCT116, where major differences can be seen. The most striking variation is the function for Raf, which is always active in all models for HCT116, meaning that the component is completely independent from the receptors and their stimulation.

This observation can be visualized in the statistical analysis of the specific pool HCT116_ps using Tremppi's tool *regulations*. When comparing the specific pool with the generic pool, the differences implied by the data can be identified. Figure 5.5 depicts this difference in a graph, where the lacking influence on Raf by its regulators EGFR, IGFR and Erk is shown. All three edges are under-represented with a frequency of 0.7, where a maximum value of 1 means that all models in the reference would contain that edge and in the subtracted pool none. Since the reference pool contains these edges too, this value is high and explained in the function $Raf = 1$ shown in Table 5.4 a. Moreover, the impact of regulators with 0.36 is high, since the maximum impact of 1 is always split among all regulators. This observation can be explained by considering the genotype of HCT116 listed in Table 1 in [54] where mutations in KRAS, RTK and PI3K are noted. KRAS is a member of the RAS family of GTP-binding proteins, which activates Raf and PI3K and is regulated by RTK. This mutation may lead to constant activation of Raf in this cell line.

Similarly, PI3K is constantly 1 in more than 70% of models of LIM1215, which again can be attributed to a mutation in KRAS present in this cell line. Note that the KRAS mutations differ between these cell lines and therefore could cause deviating effects. HCT116 though does not show a specific tendency in the regulation of PI3K, although it carries a mutation in this component.

Not all the comparisons are showing such clear differences between the pools. Table 5.4 b and c compare the pools of HCT116 with SW480 as well as SW403 with HT29 without resulting in any clear variations. For cell lines HCT116 and SW480 this could be explained by looking again at the genotypes, which show many commonly shared mutations (see Table 1 in [54]). SW403 and HT29 in c have the most similar pools of all ten comparisons, without sharing any mutation concerning components in our model. This result means that either the model is lacking a connection or a component which influences the behavior of these cell lines.

Again, we used the statistical analysis to further examine the observations from Table 5.4 c illustrated in Figure 5.6. Here, we directly computed the difference of the statistics of SW403_ps and HT29_ps represented as difference graph. As a result, the pools are exactly identical in the statistical frequency and impact of edges

upstream of Erk and Akt, although the sizes of the pools differ. Downstream of Akt and Erk there are differences, but with a very low significance in both frequency and impact.

It is interesting to note that the model pools of SW403_ps and HT29_ps are very similar, although the data sets differ in every formula but one. Thus, despite differences in genotype and measured behavior they result in very similar model pools. However, they do share an identical mutation in p53, which is a prominent tumor suppressor [104] and might govern the behavior in these cell lines.

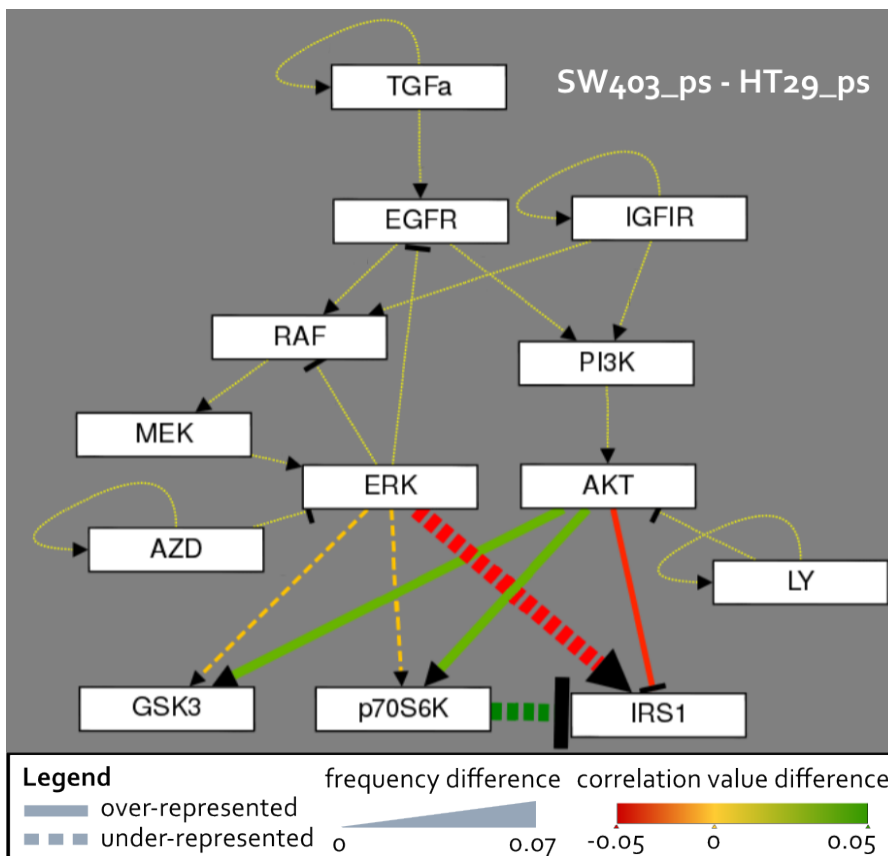


Fig. 5.6. Visualization of the statistical analysis of the differences between model pools SW403_ps and HT29_ps. The yellow dotted lines indicate that there is no difference between the statistics of the pools, meaning frequency and impact is 0.

5.2.4 Discussion

The generation and analysis of model pools using constraints that encode the available knowledge for a given system allows us to evaluate data uncertainty and guides the step from generic to more specific models. Here, we used this approach to investigate cell line specific properties of the EGFR signaling pathway. Motivated by a study of Klinger et al. [54], we first aimed at a comparison of our fully

qualitative approach with the semi-quantitative method employed in the original study. While obtaining good agreement of the results in some cases, others did not match very well. We expect that this emerges mainly from the semantics of the edge labels required by the approach.

Going beyond the results of the original study, we dropped the stability requirement for several components based on the available experimental data. By comparing the resulting model pools, we tried to find differences between cell lines and to examine whether variations in the measurements can be connected with the topology or even the genotype. Interesting insights can be derived for the cell lines LIM1215 and HCT116, where the valid regulatory functions of PI3K in LIM1215 and of Raf in HCT116 could indicate an activating mutation in that component or upstream, in these cases probably KRAS.

Observations like these are very interesting for therapeutic strategies. Often, cancer cells carry hundreds to thousands of mutations, where only few of them are causing the cancer. It is a big challenge to identify dominant players, like KRAS here, in order to decide on the most effective treatment. Here, the analysis of HCT116 cells clearly shows a cancerous rewiring of the EGFR signaling, where the component Raf is no longer dependent on growth-factor activated receptors to be activated and became insensitive to the negative feedback from Erk to terminate the signaling process.

Other comparisons, e.g. of cell lines SW403 and HT29, show only slight differences, although they do not share a mutation in components of the pathway. However, a shared mutation can be detected in the tumor suppressor p53. This protein is not directly linked to the EGFR pathway, but nevertheless might govern the behavior of these cells indirectly. For example, it was shown to influence the transcription of Pten [104], which is an inhibitor of PI3K signaling, see Figure 5.1. Also a study of Feng et al. suggests that upon stress p53 affects MAPK and PI3K signaling [28]. Thus, a model expansion by adding p53 and new p53 measurement data could help to clarify this result.

Crosstalk analysis between MAPK and PI3K signaling

In this chapter, I present an application of the crosstalk objective of the toolbox, described in Section 3.2. I was motivated to study the connections between the MAPK and the PI3K pathway by our cooperation with Christina Kuznia and Christine Sers at Charité, who investigated the treatment of renal cancer cells with the drug Sorafenib, a Raf inhibitor. They observed a different reaction upon drug treatment in different cell lines and hypothesized that the crosstalk in the cells might differ causing these variances.

For this aim, I first present the biological background on the crosstalk between MAPK and PI3K signaling. Then, the possible crosstalk in healthy cells is investigated based on literature. We published this study in [98] and present it with corrected and expanded results here, which show fairly different pool sizes, however, the interpretation is unchanged. Finally, I expanded this study adding Sorafenib to the model and analyzed the resulting models for data from renal cancer cells provided by our cooperation partner.

6.0.1 Biological background

Signaling processes orchestrate the behavior of cells, where hundreds of signals propagate through the pathways to determine its fate. Although many of these pathways have been studied extensively, the integration of these signals is less well understood [1]. Often the cell-fate depends on a whole range of signals, e.g. proliferation requires nutrients as well as growth signals, also other cell-fates overpower inputs, e.g. induction of cell death. Here, crosstalk enables a spatiotemporal control of pathways in order to integrate extracellular stimuli to distinct fates [1].

Crosstalk in cellular signaling There are four different kinds of crosstalk: negative feedback, cross-activation, cross-inhibition, and pathway convergence [69]. In detail, a negative feedback means that a component inhibits an upstream component of

its own pathway. Cross-inhibition (-activation) means that a component inhibits (activates) an upstream component of a different pathway. Lastly, pathway convergence describes the instance when two components of different pathways influence a common target [69].

In general, information on crosstalks is sparse. Since targeted therapy becomes more and more popular, this lack of knowledge poses a problem. Crosstalk can lead to resistances by redirecting the signaling to another pathway and thereby bypassing the drug target. This effect is most commonly described in the MAPK and PI3K signaling, where combined therapy approaches are often used to efficiently block the signaling [87, 77, 23].

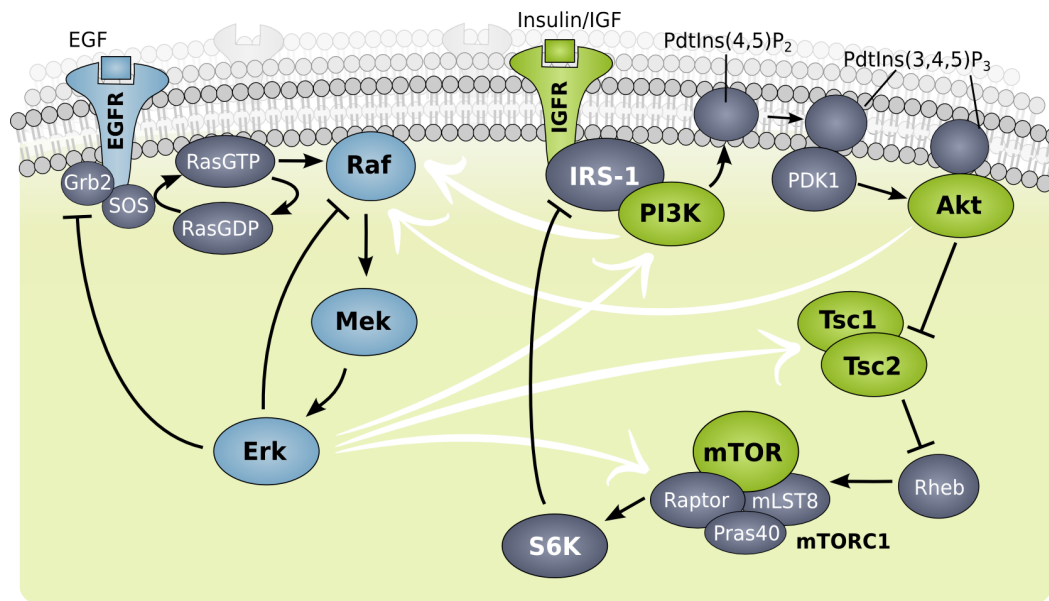


Fig. 6.1. Scheme of the MAPK and PI3K signaling cascades with crosstalk indicated by white arrows.

Crosstalk between the MAPK and PI3K pathways Based on the review of Menzoda et al., I investigated the crosstalk between MAPK and PI3K signaling. Figure 6.1 shows both pathways and potential crosstalks present in the literature. I included the cross-inhibition of Akt on Raf, of Erk on PI3K, of Erk on Tsc and the cross-activation of Erk on mTORC1 from the review [69].

In detail, strongly activated MAPK signaling was found to cross-activate PI3K signaling. Erk was observed to phosphorylate Tsc2 and thereby activate mTORC1 signaling similarly to Akt [76, 108]. Also, cells with constitutively active Ras showed phosphorylation of the mTORC1 subunit Raptor by Erk leading to an activation of the complex [10]. A cross-inhibition from Akt on Raf through phosphorylation was observed upon strong IGF stimulation [70], and cross-inhibition from Erk to PI3K was found after treatment with Mek inhibitors showing an increase in Akt

activity [114]. Additionally to the crosstalk reported in the review, I added the cross-activation from PI3K on Raf. In the study of Will et al., they found that PI3K inhibition, but not AKT inhibition, causes rapid decrease in wild type Ras activity and in Raf/Mek/Erk signaling concluding that PI3K cross-activates the MAPK cascade upstream of Ras [107].

6.1 Crosstalk analysis using literature data

In this section, I applied the objective of crosstalk analysis presented in Section 3.2 to systemically explore the interplay of the MAPK and PI3K pathway. Since this study was published [98], passages from the original paper were adopted and some additional work is shown.

Motivation As described before, both pathways are known to be connected via crosstalk, but the exact information about interactions is sparse and unclear [69]. Mutations in these pathways are very prominent in tumors, motivating research for medical purposes. Several comprehensive logical models are available and were used for studying input-output behavior [35, 80]. Since we aim at a more complex analysis under uncertainty w.r.t. the crosstalk connections, we focus for our illustration on a very much reduced representation of both MAPK and mTOR networks which is still able to reproduce the essential pathway behavior.

For this aim, I employed the workflow of the toolbox presented in Chapter 3 with the objective of crosstalk analysis. As described in Section 3.2, I started by analyzing two single models describing each pathway separately and subsequently construct an integrated model by introducing crosstalk interactions between components belonging to different isolated models. The crosstalk was labeled according to the available data. A pool of models in agreement with these constraints was then generated and further reduced to obtain model that satisfy a list of desired properties. First, I filtered for the models that preserve the validated behavior of the isolated models. Second, I incorporated new experimental data pertaining to the integrated models and analyze the model pool for commonalities and differences between the remaining models in the pool.

6.1.1 Model building and integration

System initiation The single models were built based on literature information. The MAPK model was extracted from Kholodenko et al. [49], where each compo-

ment has two states, the unphosphorylated protein is assumed to be inactive and therefore assigned to the state 0. The active, phosphorylated form is assigned to the state 1. Besides Raf, every component is regulated by only one predecessor (see Fig. 6.2 a), thus the logical function only contains one component and the sign of the connecting edge (Fig. 6.2 d). Raf is activated by RTK and inactivated by Erk through a negative feedback which terminates the signal, therefore we choose a logical AND connection.

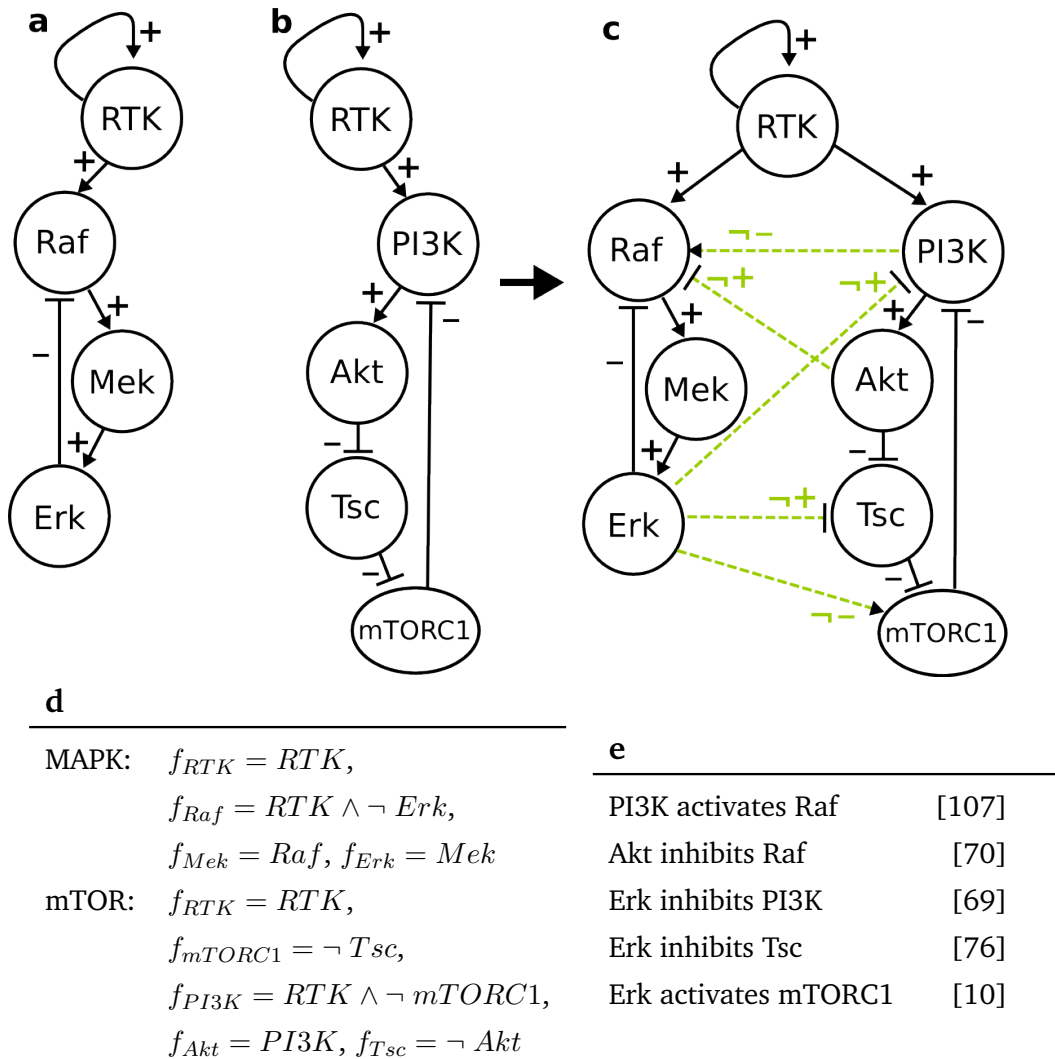


Fig. 6.2. Model setup for crosstalk analysis. **a** and **b** show the PKN of MAPK and PI3K signaling, respectively. **c** Network structure of MAPK and PI3K in black and crosstalk edges in dashed green lines with RTK as merged component. **d** Logical rules for regulations of components in single models. **e** List of crosstalk edges added to the single pathways.

The PI3K model was derived from Engelman et al. [19], where again each component is Boolean and the active state is either determined by presence of phosphorylations or the activity of a target (Fig. 6.2 b). Along with the MAPK model, the logical equations were directly determined by the predecessor and the connecting edge for

one regulator except for PI3K, where an AND gate defines the connection between the activation of RTK and the negative feedback by mTORC1 (Fig. 6.2 d). Finally, the logical rules immediately implied the edge constraints and parametrization of the components resulting in K^{MAPK} and K^{mTOR} .

Single Model Analysis Both systems exhibit a fixpoint representing a quiescent stable state and a cyclic attractor. Oscillations in agreement with the cyclic attractors were experimentally shown for the components Erk and Akt in [42], the quiescent stable state represents the behavior of the inactive pathway. Both are biologically relevant and the corresponding behavior should be preserved in an integrated model. The following properties thus make up the set \mathcal{P} :

- P_1 : \exists fixpoint in \mathcal{S}^{MAPK} with (RTK=0, Raf=0, Mek=0, Erk=0)
- P_2 : \exists attractor in \mathcal{S}^{MAPK} with RTK=1, s.t. Raf, Mek and Erk oscillate
- P_3 : \exists fixpoint in \mathcal{S}^{mTOR} with (RTK=0, PI3K=0, Akt=0, Tsc=1, mTORC1=0)
- P_4 : \exists attractor in \mathcal{S}^{mTOR} with RTK=1, s.t. Akt and mTORC1 oscillate

6.1.2 Crosstalk analysis

After defining the single model and the properties, both were integrated into one system and subsequently analyzed for literature data.

Model integration According to the second step of the crosstalk objective in Section 3.2, I integrated the MAPK and PI3K model by combining components and edges of both models (Fig. 6.2 c). Here, the component RTK featuring in both models was merged to one component, since both pathways are activated by this receptor. Lastly, the crosstalk edges were added. I selected 5 possible crosstalk connections from the literature, which are given in Figure 6.2 d. Here, the cross-activation by PI3K on Ras was transferred to Raf since Ras was not included explicitly. Based on the available biological information, they were labeled with $\neg+$ or $\neg-$, which translates to edges being either inhibiting resp. activating or not functional.

The parametrizations of components not targeted by any crosstalk edge were determined by the single models. The other components, namely RTK, Raf, PI3K, Tsc and mTORC1, have new regulatory contexts. For them, I considered all parametrizations in agreement with the edge labels, as defined in Section 2.4, leading to a model pool of size 13,266.

In the next step, the properties $P_1 - P_4$ observed in the single models needed to be transferred to the dimension of the coupled system.

- P_1 and P_3 both characterize the steady state without stimulus and are fused to one property: **FP1**: \exists fixpoint in \mathcal{S} with (RTK=0, Raf=0, Mek=0, Erk=0).
- P_2 and P_4 both describe cyclic attractors, assumed to be preserved in MAPK and mTOR components as 2 distinct attractors in the state space:
 $P_2 \rightarrow$ **Cyc.MAPK**: \exists attractor in \mathcal{S} where Raf, Mek and Erk oscillate,
 $P_4 \rightarrow$ **Cyc.mTOR**: \exists attractor in \mathcal{S} where Akt and mTORC1 oscillate.

To obtain the set \mathcal{K} of parametrizations satisfying the properties in \mathcal{P} we used TomClass. The exact specification is given in Table 6.1 **b**.

Tab. 6.1. CTL formulas for filtering model pool derived from properties of the single models and experimental data.

a	BAY			MK2206				read-out
	Time [h]	0	0.5	2	0	0.5	2	
P-Akt	1	0	0	1	0	0	0	Akt
P-S6	1	1	0	1	1	0	1	mTORC1
P-Erk	1	0	1	1	1	1	1	Erk

b

FP1: EF(Delta=0&Raf=0&Mek=0&Erk=0&mTORC1=0) IS:RTK=0

Cyc.MAPK: EF(AG(EF(deltaErk!=0))) IS:RTK=1

Cyc.mTOR: EF(AG(EF(deltamTORC1!=0)&EF(deltaAkt!=0))) IS:RTK=1

FP2: EF(Delta=0&Erk=1&Akt=1) IS:/ Fixed:RTK=1,PI3K=1

MK: EF(mTORC1=1&Erk=1&EF(mTORC1=0&Erk=1&EF(mTORC1=1&Erk=1)))
IS:mTORC1=1,Erk=1 Fixed:Akt=0,RTK=1,PI3K=1

BAY: EF(Akt=0&mTORC1=1&Erk=0&EF(Akt=0&mTORC1=0&Erk=1))
IS:Akt=1,mTORC1=1,Erk=1 Fixed:RTK=1,PI3K=0

a Table with discretized western blot data from Will et al. [107] for PI3K inhibitor BAY 80-6946 and Akt inhibitor MK2206. **b** CTL formulas for properties **FP1**, **Cyc.MAPK**, and **Cyc.mTOR** as well as **FP2**, **MK**, and **BAY** derived from western blot data.

Data formalization To reduce the pool further, experimental data from literature containing information about the integrated system was used. A study from Will et al. investigated the effect of Akt and PI3K inhibitors in connection with MAPK signaling in a breast cancer cell line [107]. The genotype of the cell line needs to be considered when exploiting cancer data. This specific cell line (BT-474) carries an amplification in HER2, which belongs to the RTK family and a mutation in PI3K (PIK3CA) which causes increased levels of activity. As described in Section 3.3, we added the genotype information which amounts to adding the `fixed` component constraint to the properties derived from the corresponding experimental observations, fixing

RTK and PI3K to value 1 unless explicitly indicated otherwise in the experimental set up.

In the paper, time series experiments without inhibitor were performed (see Fig. S2 B in [107]), indicating that the quiescence state shows active Akt (P-Akt) and active Erk (P-Erk). In contrast to **FP1** the mutations cause a different quiescence state than in the wild type, thus there is no conflict between these observations. The data was translated into a CTL formula **FP2** shown in Table 6.1 c. Moreover, Will et al. performed measurements perturbing with Akt inhibitor MK2206 and PI3K inhibitor BAY 80-694 hypothesizing that PI3K is upstream of MAPK and blocking this kinase should affect Erk activity. I used these western blots (shown in Fig. 2 in [107]) to further refine the model pool. In order to avoid discretization errors, time points with unambiguously active or inactive states were chosen, discretized and collected in Table 6.1 a. The table shows the states of Akt, Erk and P-S6, which is a kinase dependent on mTORC1 and therefore used as its read-out. For the PI3K inhibitor three measurements and for the Akt inhibitor four measurements were implemented as CTL formulas **BAY** and **MK**, listed in Table 6.1 c.

Choice of strictness The model checker TomClass has parameters for how strict a CTL formula can be applied, as described in Sections 2.7 and 3.3. In order to explore the influence of this parameter of the filtering effect on the pool, I tested every formula with both options for initial states: `ForAll` and `ForSome` (see Table 6.2). As a result, a general difference for CTL formulas encoding an attractor such as **FP1**, **Cyc.MAPK**, **Cyc.mTor**, and **FP2**, where all model pools were identical except for **FP1** showing a slightly smaller pool size for the parameter `ForSome`.

This observation can be explained looking at the structure of the models, where RTK is the only input to the system. We know that the number of attractors is limited by the number of inputs, except for systems with positive feedbacks which can lead to multistattonarity [100]. Here, the single models contain one negative feedback each, but no positive. Also, adding the crosstalk to the system only adds negative edges, which in combination with existing negative feedbacks rarely lead to positive feedbacks in this system. Using the `annotate_attractors` function in TomClass (see Sec. 2.7.1), I determined how many models have multistattonarity for $RTK = 0$ and $RTK = 1$, which was 114 in each case. This means that less than 1% of the models express multistattonarity. Thus, most models have a maximum of 2 attractors, one for $RTK = 0$ and one for $RTK = 1$.

Tab. 6.2. Filtering the generic pool for properties and data sets

	FP1	Cyc.MAPK	Cyc.mTor	FP2	MK	BAY
ForSome	4489	7062	5886	6633	10318	13266
ForAll	4396	7062	5886	6633	770	10318

Specific pool sizes filtering for CTL formulas each show the effects of the strictness parameter. The formulas testing attractors are less sensitive to this parameter showing the same or similar pool sizes. For time series measurements strong variations are observed.

The cyclic attractors **Cyc.MAPK** and **Cyc.mTor** both show the same pool size for each parameter setting, see Table 6.2. By annotating the attractors I observed that multistatony models do not show cyclic attractors. The positive cycle, which requires an edge from Akt to Raf and a pairwise combination of the edges {(Erk, Tsc), (Erk, mTORC1), (Erk, PI3K)} seems to interfere with the capability to show a cyclic attractor, although there are intact negative cycles present. The measured fixpoint **FP2** also does not show any models with positive cycles, thus all initial states reach the same steady state and there is no difference between the parameter settings.

However, the trivial fixpoint **FP1** poses an exception, where some models are valid for *ForSome* but not for *ForAll*, because the CTL formula is valid for models that carry a positive cycle having two fixpoints. In such a case, *ForSome* is true if one fixpoint is valid and *ForAll* require both fixpoints to agree with the formula. Since this fixpoint represents the inactive state of the signaling system, which should be inactive without an active receptor, we want to exclude multistatony and choose the parameter *ForAll* for **FP1** for our further analysis,

The time series data sets **MK** and **BAY** show a clear difference in the pool size (Tab. 6.2). Especially the CTL formula **MK** shows more than 10 times more models for the low strictness. This fact is surprising, since in the data set five out of eight components are defined in the initial state, which is the highest number over all formulas. Therefore, the difference in the set of states between *ForSome* and *ForAll* should be the smallest. Moreover, the strict version of **MK** is in conflict with **FP2** leading to an empty pool for intersection. In conclusion, for the time series data we choose the low strictness.

Model Pool Analysis After filtering, the resulting pool contains 554 models that are in agreement with the properties derived from the single networks and the experimentally observed behavior. I employed the statistical analysis approach presented in Section 3.2 to investigate the topological characteristics of the integrated models. For identifying important influences and structures in the model pool, the

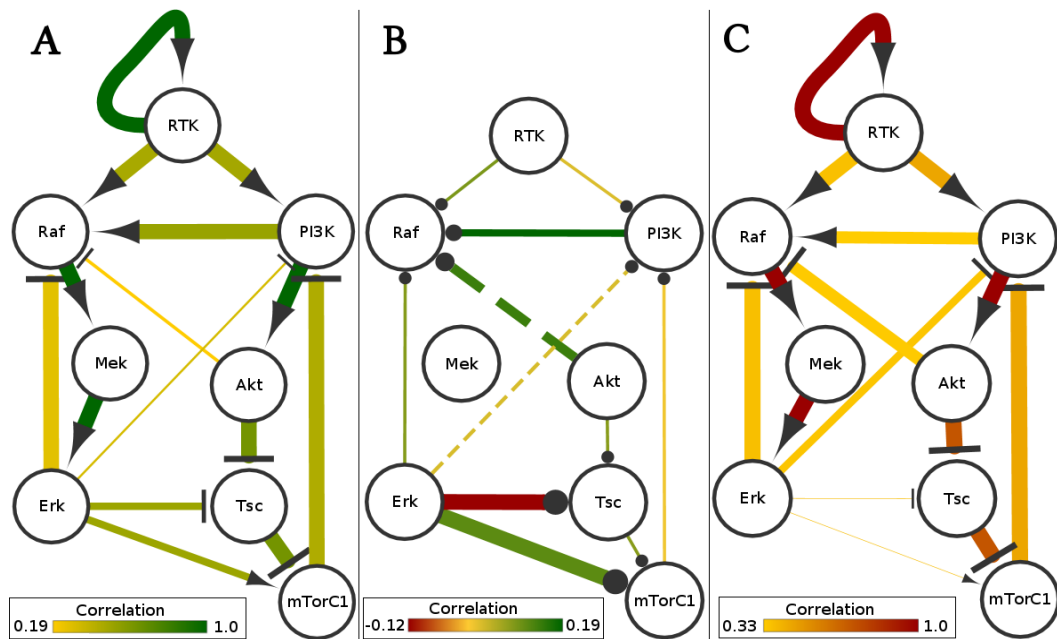


Fig. 6.3. Visualization of the statistical analysis of the model pool. The edge width represents the frequency of occurrence in the pool and the heads show the sign. **A** illustrates the refined pool $\mathcal{K}_{\mathcal{R}}$, and **C** the reference pool \mathcal{K}_{ref} . In **B** the difference graph is shown, where the edge signs has been dropped, solid lines are drawn for connections more frequent on average in $\mathcal{K}_{\mathcal{R}}$ than in \mathcal{K}_{ref} and dashed lines for lower frequency in the refined pool. Figures generated using Cytoscape (<http://www.cytoscape.org/>).

frequency of the edges and correlations between the components is calculated. In Figure 6.3 the statistical analysis of the refined model pool $\mathcal{K}_{\mathcal{R}}$ resulting after the filtering as well as a reference pool \mathcal{K}_{ref} , and the difference between the two is visualized. The reference pool contains all models of the originally generated pool that have been discarded in the filtering process.

The result for $\mathcal{K}_{\mathcal{R}}$ is shown in Figure 6.3 **A**, where the crosstalk edge from PI3K to Raf has a frequency of 1. Thus, in every model in the pool this edge is functional. In order to evaluate possible enrichments of other edges, we need information about \mathcal{K}_{ref} , shown in Figure 6.3 **C**. Here, the frequency and correlation is given by the edge constraints and the arising combinations of parametrizations. Finally, the difference between the filtered and the reference pool is depicted in **B**. Again, the connection PI3K and Raf is shown to be more prominent and highly correlated comparing the filtered models to all possible models. This result is in line with findings of Will et al., where it was concluded that PI3K is upstream of the MAPK cascade [107]. The crosstalk from Erk to PI3K is less frequent in $\mathcal{K}_{\mathcal{R}}$, which is reasonable since in the data PI3K is a fixed component and therefore this edge cannot be functional here.

Moreover, the influence of Erk on Tsc and mTORC1 is strongly enriched in the selected model pool. Applying an exact analysis by classification of the pool (see

Tab. 6.3. Classification for crosstalk edges and CTL formulas

Crosstalk	(PI3K, Raf)	(Akt, Raf)	(Erk, mTOR)	(Erk, Tsc)	(Erk, PI3K)	Size
2	1	0	0	1	0	3
2	1	0	1	0	0	6
3	1	0	0	1	1	6
3	1	0	1	0	1	17
3	1	0	1	1	0	12
3	1	1	0	1	0	9
3	1	1	1	0	0	27
4	1	0	1	1	1	39
4	1	1	0	1	1	18
4	1	1	1	0	1	123
4	1	1	1	1	0	45
5	1	1	1	1	1	249

Classification of $\mathcal{K}_{\mathcal{R}}$ shows the essential edge from PI3K to Raf and an essential influence from Erk to mTORC1 either directly or via Tsc. Crosstalk counts the number of present crosstalk edges and Size gives the number of models in the class.

Section 2.7), showed that every model in $\mathcal{K}_{\mathcal{R}}$ contains at least one of these edges shown in Table 6.3. Here, the crosstalk edges and the CTL formulas were used as features to group models. These groups only differ in their parametrizations, but show the same behavior towards the tested data.

Interestingly, the analysis shows that a minimum of 2 edges are required to explain the data. Moreover, none of the edges is rejected by the data, but the connection from Akt to Raf and Erk to PI3K are only possible in combination with the essential edges.

6.1.3 Discussion

Biological processes do not work isolated, but in concert with other cellular mechanisms. For many of these processes there exist validated models, but their interactions among each other are often unclear. Here, I applied the workflow presented in Chapter 3 with the objective to investigate crosstalk. This workflow first investigates the properties of the single models that are required to be preserved in the integrated system. As validation that model integration does not change the dynamics of both pathways, I found that the uncoupled system, where all crosstalks are absent, is in agreement with all properties from the single models **FP1**, **Cyc.MAPK** and **Cyc.mTOR**.

In the next step, I reduced the pool as much as possible and performed the pool analysis. Another possibility would be to add constraints stepwise and perform analysis after each step. This may allow to link specific network characteristics to the properties and functionalities encoded in each constraint. However, I was able to extract two essential crosstalks where the connection from PI3K to Raf is already known in the literature. The influence from Erk to mTORC1 directly or through Tsc is more interesting and an experiment to dissect these parallel influences would further restrict the pool.

The analysis using different levels of strictness revealed that the influence of *ForSome* and *ForAll* differs among the data sets. Attractors showed almost no difference in the model pool sizes, whereas the time-series data was much more affected. One explanation is that most models only have one attractor for each input state, thus if one trajectory is valid, all of them are. The time-series data also contained more constraints on components by having a sequence of states, which might be harder to fulfill than reaching an attractor.

6.2 Signaling in RCC cells: role of Sorafenib and crosstalk

This study is a collaboration project with Christine Sers and Christina Kuznia from the institute of pathology at Charité Berlin. In her PhD studies, Christina Kuznia investigated the signaling processes in several renal cell carcinoma (RCC) cell lines, where these supposedly similar cancer cells behaved very differently towards stimulation and drug treatment [55]. As a consequence, the signaling processes involved in the cell fate decisions, such as survival or apoptosis, were examined and analyzed.

One focus of these studies were the MAPK and PI3K pathway as well as their crosstalk. Since these processes incorporate uncertain mechanisms such as the crosstalk and the effect of mutations within the pathway, I wanted to apply our modeling approach to support the investigation. This study includes all three objectives formulated in Chapter 3: a crosstalk analysis, testing of a drug, and examining the effect of a mutation.

Sorafenib Sorafenib is a cancer drug developed to inhibit pathways controlling proliferation and cell survival like MAPK and PI3K cascades with anti tumor activity in colon, breast and non-small lung cancer [106]. The multikinase inhibitor Sorafenib

was developed to suppress the MAPK pathway, since this pathway described in function of both a tumor suppressor as well as pro-oncogenic [9]. Though it was initially found to target Raf, Sorafenib influences a wide variety of receptor tyrosine kinases (RTKs) [106]. Very recently, Sorafenib was also shown to inhibit the IGFR in vitro [111].

Applying Sorafenib in experiments to different RCC cell lines, we observed that these cells undergo apoptosis, but the cell lines reacted on very different time scales and with variations in their signaling behavior [55]. For clarifying these differences, Kuznia et al. performed more experiments using inhibitors that are assumed to act on the same targets, but these inhibitors were not able to induce apoptosis. So far it is not completely clear how Sorafenib acts on the MAPK and PI3K pathways.

6.2.1 Model definition and objective formalization

System initialization For the investigation of the different RCC cell lines, the crosstalk between MAPK and PI3K as described in the previous section is of interest. Additionally, I wanted to test whether Sorafenib acts on its designated target Raf or whether it might act through EGFR or IGFR in these cells. For this aim, I split the component RTK into two components with distinct activators, their ligands EGF and IGF respectively. MAPK is the canonical pathway activated by EGFR and PI3K signaling is the main effect of IGFR activation. Although the receptor was split, the ability of cross-activation of the non-canonical pathway is preserved for IGFR on MAPK through the crosstalk from PI3K on Raf. For the crosstalk of EGFR on PI3K signaling there is a possible cross-activation through Ras which is downstream of EGFR and upstream of Raf [109] (see Fig. 6.1). We added this interaction as additional crosstalk edge from EGFR on PI3K.

For the analysis, biological information from prior experiments about the RCC cells was incorporated. First, in one cell line mTOR was found to carry a mutation, whose effect is unknown. To account for this mutation all outgoing edges of mTORC1 were set to optional in the PKN, which affects the feedback from mTORC1 to IGFR. Secondly, the VEGFR was not included in the model, since it was found to not be expressed in the later described experiment, although it is a major target of Sorafenib.

Finally, the results of the previous study on crosstalk between MAPK and PI3K in healthy cells [98] were included into the PKN of this model. In detail, the candidate edges from Akt to Raf and Erk to PI3K were excluded, since they were under-

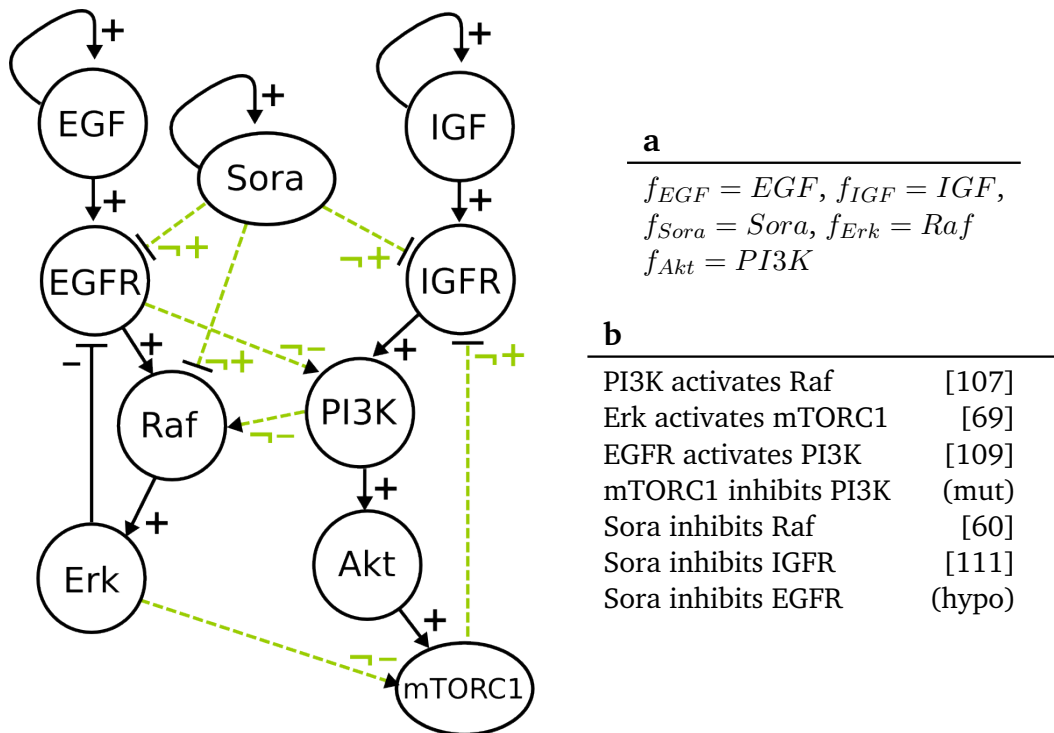


Fig. 6.4. Interaction graph of the MAPK (left hand side) and PI3K (right hand side) model marked with solid lines and optional influence of Sorafenib and crosstalk marked with dashed lines. **a** Predefined logical rules for regulations of components without optional incoming edges. **b** List of optional edges added to the network with references.

represented in the results. The essential influences were preserved as optional edges, with PI3K as Raf activator and Erk as mTORC1 activator summarizing both the direct edge and the indirect influence via Tsc. Since we had no data to further clarify this influence, this simplification reduced the complexity of the study without changing the logical structure of the model.

Moreover, components that were neither measured nor perturbed are excluded from the model to reduce the complexity. Here, Mek and Tsc were not considered in the model, since both were lined up in a cascade as components with single input and output, thus deleting them does not pose problems for the model dynamics.

Cell line specific crosstalk analysis After presenting a systematic approach to integrate the MAPK and PI3K pathway via crosstalk for healthy cells in the previous section, I wanted to explore the cell line specific crosstalk and the effect of Sorafenib in this study. For this aim, the model was fit to the new data and Sorafenib is added as additional input to the system, where I wanted to test Raf, EGFR and IGFR as possible targets. Moreover, one cell line, MZ1851RC carries a mutation in mTOR with unknown effect for mTORC1, thus the outgoing edge to IGFR was set to optional. Therefore, this study combines all three presented objectives described

in Section 3.2: the crosstalk analysis, identifying driver mutations and testing the effect of drugs.

6.2.2 Data processing & formalization

Our collaboration partner examined the signaling mechanisms of renal cancer cell lines MZ1257RC, MZ1851RC and MZ1795RC. The work was based on the observation that some cells were sensitive to a treatment with Sorafenib while others were resistant [55]. For our investigation, we used two different data sets: Western blot measurements of mTORC1 activity over time and a high throughput assay. Here, we restrict the analysis to two cell lines MZ1257RC and MZ1851RC, since there was no assay for cell line MZ1795RC available.

Western blot measurements of mTORC1 targets In the western blot measurements done by Christina Kuznia, the activity of mTORC1 was measured by its targets p70S6K (S6K), S6RP, and 4E-BP1 in MZ1257RC and MZ1851RC cells (see Fig. 6.5). Here, the cells were either treated with DMSO or Sorafenib and the phosphorylation of the mTORC1 target was measured over time. Regarding the measurements until 12 hr, MZ1257RC cells showed a significant decrease in phosphorylation levels for S6K and S6RP. However, MZ1851RC cells only showed a reduction in S6RP phosphorylation for later time points, but the phosphorylation of S6K remained high. I did not consider the 24 h time point, since we are only interested in signaling effects and this measurement is likely to be influenced by transcriptional effects.

The phosphorylation of the mTORC1 target 4E-BP1 is on the same level for both cell lines, for the treated and the untreated control sample. Since our model only represents mTORC1 in the function as inhibitor of IGFR by the negative feedback through S6K, I did not consider the measurements of 4E-BP for our study. Instead, S6K was used as the read-out for the mTORC1 activity in the formal encoding of the Western blot data as CTL formulas **DMSO**, **Sora1257** and **Sora1851** in Table 6.5. Here, both cell lines show active mTORC1 for DMSO treatment throughout the measurements, thus a steady state was assumed. For Sorafenib treatment, cell line MZ1257RC shows a steady state with decreased S6K phosphorylation, thus mTORC1 was set to 0. In contrast, cell line MZ1851RC has stable S6K phosphorylation, thus mTORC1 was set to 1.

Bio-Plex® experiments for a more detailed view on pathways After observing differences in the activity of mTORC1 in the Western blots towards Sorafenib treatment, we wanted to investigate where the differences in the upstream regulation of mTORC1

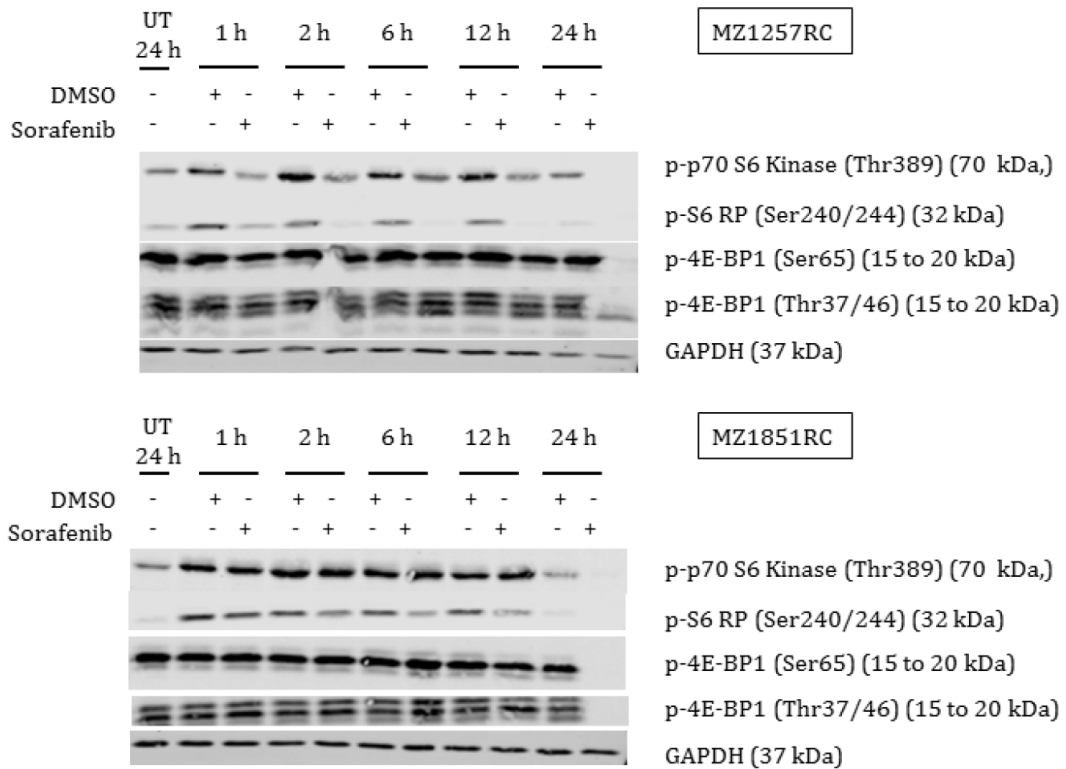


Fig. 6.5. mTORC1 activity was inhibited in MZ1257RC cells due to Sorafenib as indicated by a decrease in phosphorylation level of S6K and S6RP, but phosphorylated 4E-BP1 remained stable. MZ1851RC analysis revealed only a weak inhibition of S6RP phosphorylation, but not of p70S6K. Figure courtesy of [55].

originate from. For this aim, a throughput approach using the Bio-Plex[®] system was applied [55], which is a Luminex assay as described in Chapter 5. Here, the cells were unstimulated and not starved but treated with Sorafenib or DMSO and measured at different time points over a total period of 36 hours. The experiments, measurements and preprocessing of the data was done by Christina Kuznia [55].

In detail, the activity of the PI3K/mTORC1 signaling pathway was measured by the phosphorylation of Akt, mTOR, p70S6K, S6RP, and Pten. The MAPK activity was determined through the phosphorylation of Erk and p90RSK. Moreover, the receptors EGFR, IGFR and VEGFR were included into the experiment, since we were interested which receptor is targeted by Sorafenib and to account for the feedback processes. The complete dataset provided by [55] is listed in the Supplement (Fig. A.6).

For the analysis, the first preprocessing step was a reduction of the data set, shown in Table 6.4. The components that are not present in the model were not considered in our analysis (p90RSK and Pten). The measurements of the VEGFR showed

very low levels with only very small deviations, suggesting that this receptor is not expressed in the cells. The antibody for S6RB delivered poor results, same accounts for Akt pT308 antibody, thus only the results for Akt pS473 was used. Moreover, the mTOR phosphorylation site is not specific for its activity [65], thus I used the phosphorylation of the component p70S6K as readout for mTORC1.

In the next step, I discretized the data by defining a threshold for each component. For this aim, the arithmetic mean of every component within one cell line but for both treatments was calculated. Since the cells were cultivated in parallel and treated in one experiment, I expected the phosphorylated levels for both treatments to be comparable. Thus, the mean value for e.g. Erk has the same mean value under both Sorafenib and DMSO treatment. Moreover, the standard deviation for each component was calculated in order to avoid the problem of discretizing a data set, where components do not change over time. By looking at small standard deviations relative to the mean, I identified the IGFR measurements in MZ1257RC as a problematic component (see Table 6.4). Comparing the IGFR levels between the cell lines, we decided to exclude the data for this component in the cell line MZ1257RC, since the measurements suggest a constant behavior of the receptor.

Since we are interested in the signaling processes, I only included measurements until 8 hours. The resulting CTL formulas are listed in Table 6.5, where for **Bp1851Sora**, **Bp1851DMSO**, and **Bp1257Sora** each measurement was encoded as transient state and **Bp1257DMSO** was encoded as steady state, since there was no change in the discretized data over time (see Table 6.4 MZ1257RC-DMSO). Moreover, as choice of strictness I chose the most relaxed option *ForSome*.

6.2.3 Analysis of cell line specific model pools

The model pool resulting from combinations of optional edges contains 19,404 models. In order to find biologically relevant models, I filtered this pool for those models that are able to simulate experimentally observed behavior. Unlike the crosstalk study in Section 6.1, I dropped the condition of a quiescence state and oscillations in MAPK and PI3K as general properties, since we are looking at cancer cells.

Cell line specific model pools Each CTL formula has a non-zero pool size and is therefore feasible for our analysis (see Table 6.5). To determine the cell line specific models, I calculated the intersection of the different subpools as described in Table 6.5 for cell line MZ1257RC as Rp.1257 and for cell line MZ1851RC as Rp.1851.

Tab. 6.4. Reduced data set of Bio-Plex® experiments for the cell lines MZ1851RC and MZ1257RC

	MZ1851RC-Sora					MZ1851RC-DMSO				
	Erk	EGFR	S6K	Akt	IGFR	Erk	EGFR	S6K	Akt	IGFR
20m	307.5	118	230	540.5	176	737	223	498.5	2468.5	221
40m	460.5	169	344	1941	210	664	223	449.5	2572	227.5
1h	518	200	386.5	2504	247	601	161	328	1850.5	178
1.5h	452	160	327.5	2289	195.5	508	202	329	1658	202
2h	455.5	146	319	1861	223	439	156	296	1552	162
4h	255	101	205	820	181.5	350	127	236	874	135
8h	232	84	161	517.5	181	221.5	102.5	185	476	137
12h	200.5	70	152	623.5	139	217	91.5	184	600	117
24h	22	42	16	39	59	84	60	80	879.5	90
36h	23.5	40.5	15	38	52	61	53	72	286	125
con	65	48	70	200	165	65	48	70	200	165
Mean	315	119	225	1127	163	315	119	225	1127	163
StD	217	61	139	873	53	217	61	139	873	53
20m	0	0	1	0	1	1	1	1	1	1
40m	1	1	1	1	1	1	1	1	1	1
1h	1	1	1	1	1	1	1	1	1	1
1.5h	1	1	1	1	1	1	1	1	1	0
2h	1	1	1	1	1	1	1	1	1	0
4h	0	0	0	0	1	1	1	1	0	0
8h	0	0	0	0	1	0	0	0	0	0

	MZ1257RC-Sora					MZ1257RC-DMSO				
	Erk	EGFR	S6K	Akt	IGFR	Erk	EGFR	S6K	Akt	IGFR
20m	353.5	136	278	569	48	491	178.5	355	659	58
40m	157	81	149	444.5	49	383.5	143.5	305	787	51
1h	160.5	83	142	365	44	386	138.5	308	665	49
1.5h	204	128	231	430	50	387	155	350	817.5	48
2h	253	160	290	557	46	379	152.5	357.5	847	51
4h	196	139	205	411	44	446.5	188.5	424	973	49
8h	288.5	224	335	434	55	346	153.5	313	468	56
12h	67	67	94.5	115	51	264	137.5	279	449.5	48
24h	73	67	84	345.5	53	168	103	176	411	44
36h	39.5	56	49	82	48	156	127	214.5	169.5	42
con	92	54	117.5	86.5	45	92	54	117.5	86.5	45
Mean	233	121	230	453	48	233	121	230	453	48
StD	136	47	106	259	4	136	47	106	259	4
20m	1	1	1	1		1	1	1	1	
40m	0	0	0	0		1	1	1	1	
1h	0	0	0	0		1	1	1	1	
1.5h	0	1	1	0		1	1	1	1	
2h	1	1	1	1		1	1	1	1	
4h	0	1	0	0		1	1	1	1	
8h	1	1	1	0		1	1	1	1	

Time series measurements for both cell lines with DMSO or Sorafenib treatment are shown from 20 minutes (m) to 36 hours (h) and an untreated control sample (con). For the calculating the arithmetic mean and standard deviation (StD), the data from both treatments within each cell lines was processed together and all time points were considered. The data is discretized using the mean as threshold value.

Tab. 6.5. Filtering model pool using model checking.

CTL formula	Pool size
DMSO: EF(AG(mTORC1=1)) IS:Sora=0	15,026
Sora1257: EF(AG(mTORC1=0)) IS:Sora=1	15,026
Sora1851: EF(AG(mTORC1=1)) IS:Sora=1	5,902
Bp1851Sora: EF(mTor=1&Akt=0&EGFR=0&Erk=0&IGFR=0&EF(mTor=1&Akt=1&EGFR=1&Erk=1&IGFR=1&EF(mTor=0&Akt=0&EGFR=0&Erk=0&IGFR=1))) IS:Sora=1	9,624
Bp1851DMSO: EF(mTor=1&Akt=1&EGFR=1&Erk=1&IGFR=1&EF(mTor=1&Akt=1&EGFR=1&Erk=1&IGFR=0&EF(mTor=1&Akt=0&EGFR=1&Erk=1&IGFR=0&EF(mTor=0&Akt=0&EGFR=0&Erk=0&IGFR=0)))) IS:Sora=0	10,080
Bp1257Sora: EF(mTor=1&Akt=1&EGFR=1&Erk=1&EF(mTor=01&Akt=0&EGFR=0&Erk=0&EF(mTor=1&Akt=0&EGFR=1&Erk=0&EF(mTor=1&Akt=1&EGFR=1&Erk=1&EF(mTor=0&Akt=0&EGFR=1&Erk=0&EF(mTor=1&Akt=0&EGFR=1&Erk=1)))) IS:Sora=1	9,984
Bp1257DMSO: EF(Delta=0&mTor=1&Akt=1&EGFR=1&Erk=1) IS:Sora=0	12,096
Rp.1257 = DMSO \cap Sora1257 \cap Bp1257Sora \cap Bp1257DMSO	3,658
Rp.1851 = DMSO \cap Sora1851 \cap Bp1851Sora \cap Bp1851DMSO	881

CTL formulas derived from Western blot and Bio-Plex[®] experiments and pool size gives the number of models in agreement. Rp.1257 and Rp.1851 are the cell line specific pools as the intersection of the respective data sets.

Note that both pools are required to fulfill **DMSO**, since this dataset was identical for both cell lines.

Although the single CTL formulas resulted in relatively large pools, most containing around 10,000 models or more, the intersection for the cell line specific pools shows a strong reduction with 3,658 models for Rp.1257 and 881 models for Rp.1851. Thus, there exists a cell line specific pool for each cell line, which was analyzed for information on crosstalk and Sorafenib targets by classifying both pools for the optional edges. The full tables are given in the Supplement in Figures A.7 and A.8 for MZ1257RC and Figures A.9 and A.10 for MZ1851RC.

In general, for both specific pools there are no rejected or essential edges, since each edge appears in some model but not in all of them. Here, both pools contain at least one crosstalk edge, but interestingly both pools have valid models without

any influence of Sorafenib. However, no Sorafenib target was rejected for both cell lines.

Exact analysis shows minimal mechanisms Since we are interested in possible Sorafenib targets in the system, I analyzed the cell line specific pools for two features: the number of Sorafenib targets and possible target-crosstalk mechanisms. Due to the large number of models in the pools, I separated the pool for three scenarios: no influence of Sorafenib, meaning that all three optional outgoing edges of Sora are not present, Sorafenib has one target only, Sorafenib has exactly two targets and Sorafenib has exactly three targets. In Table 6.6, the minimal models according to these scenarios for the pool Rp.1257 and in Table 6.7 for the pool Rp.1851 are listed.

The specific pool for MZ1257RC shows five models without any connections of Sora to the signaling pathways. Here, the minimal models contain either the crosstalk (EGFR, PI3K) or a connection (Erk, mTORC1) and (PI3K, Raf) (see Tab. 6.6 a). All further models are combinations of these edges. Furthermore, none of the models contains the feedback from mTORC1 to IGFR. Also more complex models with more than two crosstalk edges are not present in the pool, which is surprising since models with more edges tend to fit data more easily.

In case Sorafenib only affects one target in the model, 174 models were filtered from Rp.1257. Here, all three targets are possible with at least one extra edge. The minimal models are listed in Table 6.6 b, where IGFR as Sorafenib target either requires the feedback from mTORC1 or the crosstalk (EGFR, PI3K). For EGFR as target, an influence on mTORC1 either through crosstalk by (EGFR, PI3K) or by (ERK, mTORC1) directly is necessary. Lastly, for Raf as Sorafenib target only one mechanism with one crosstalk edge is possible, which contains an edge from EGFR on PI3K. In the presented minimal models, the edge (PI3K, Raf) is not present, however, this edge only appears in combination with the minimal models (see Supplement Fig. A.7 and A.8).

In the third scenario, Sorafenib has two functional outgoing edges and represents the largest group of models in Rp.1257 with 1,350 models. Here, every combination of the targets EGFR, IGFR and Raf is present, but only in combination with crosstalk edges. For the combination EGFR/IGFR, a minimal model with each optional edge exists except for the cross-activation from PI3K on Raf (Table 6.6 c). The combination Raf/IGFR requires either the feedback (mTORC1, IGFR) or the activation through (PI3K, EGFR) is required. Lastly, the combination Raf/EGFR requires either the

Tab. 6.6. Classification for crosstalk edges and CTL formulas for Rp.1257

	Sora targets	(EGFR, PI3K)	(Erk, mTORC1)	(mTor, IGFR)	(PI3K, Raf)
a	None	1	0	0	0
		0	1	0	1
b	IGFR	0	0	1	0
		1	0	0	0
	EGFR	0	1	0	0
		1	0	0	0
	Raf	1	0	0	0
c	IGFR/EGFR	0	0	1	0
		0	1	0	0
		1	0	0	0
	IGFR/Raf	0	0	1	0
		1	0	0	0
	EGFR/Raf	0	1	0	0
		1	0	0	0
d	All	0	0	1	0
		0	1	0	0
		1	0	0	0

Minimal models after classification of Rp.1257 for **a** no Sorafenib targets, **b** exactly one target, **c** exactly two targets, and **d** all three possible targets are affected.

crosstalk (EGFR, PI3K) or (Erk, mTORC1). Along with the minimal models for one target, the crosstalk from PI3K to Raf is only present in combination with minimal models.

Finally, for the case that Sorafenib affects all three targets IGFR, EGFR and Raf, there are 1,220 model in in Rp.1257. Similarly to the minimal models with less targets, the crosstalk edge from PI3K to Raf is not present, but there are models for all other optional edges present (Tab. 6.6 **d**).

The second cell line MZ1851RC yielded a much smaller pool Rp.1851 than Rp.1257 with 881 models. However, the proportion of models without Sorafenib influence is much higher with more than 7% in comparison to around 0.1%. The minimal models shown in Table 6.7 **a**, require either the feedback from mTORC1 to IGFR or the crosstalk from EGFR to PI3K to be present. The other crosstalk edges appear as combinations with either of them or both.

In the scenario of Sorafenib affecting one target, all three options are possible. Looking at the minimal models, EGFR and Raf only require a single additional edge to be present, while IGFR requires two (Tab. 6.7 **b**). Here, for models having EGFR

Tab. 6.7. Classification for crosstalk edges and CTL formulas for Rp.1851

	Sora targets	(EGFR, PI3K)	(Erk, mTORC1)	(mTor, IGFR)	(PI3K, Raf)
a	None	0	0	1	0
		1	0	0	0
b	IGFR	0	1	1	0
		1	0	1	0
		1	1	0	0
	EGFR Raf	0	0	1	0
		0	0	1	0
		1	0	0	0
c	IGFR/EGFR				
		IGFR/Raf	0	1	1
	EGFR/Raf	1	0	1	0
		1	1	0	0
		0	0	1	0
		1	0	0	0
d	All	1	0	1	0

Minimal models after classification of Rp.1851 for **a** no Sorafenib targets, **b** exactly one target, **c** exactly two targets, and **d** all three possible targets are affected.

as Sora target the negative feedback from mTORC1 on PI3K is essential. Models with Raf as target either contain the cross-activation from EGFR on PI3K or the feedback. For IGFR, any pairwise combination of optional edges except for (PI3K, Raf) appears. Like in cell line MZ1257RC, this edge is not required to be present, but can be active in combination with other edges.

In case Sorafenib acts on two targets, 417 models in Rp.1851 are filtered. Surprisingly, not every combination is possible in this selection since there is no model with IGFR and EGFR as Sora targets. The combination Raf/EGFR requires one additional edge to be functional, which are the same as for Raf only (Tab. 6.7 **c**). For the combination IGFR/Raf two additional edges are necessary, which in this case matches the edges from IGFR only.

Finally, if we expect Sorafenib to act on all three targets, only 96 models are valid. These models all require at least two crosstalk edges to be present, which are the negative feedback and the cross-activation by EGFR on PI3K. All other edges appear in combinations with the minimal model shown in Table 6.7 **d**.

6.2.4 Discussion

Two RCC cell lines, MZ1257RC and MZ1851RC, were observed to behave differently upon Sorafenib treatment. The project was supposed to examine possible targets of Sorafenib in the MAPK and PI3K signaling and also investigate the uncertain crosstalk between these pathways by testing a generic model for data from both cell lines. Although there was a substantial reduction from the initial pool with 19,404 to 3,658 for MZ1257RC and even 881 for MZ1851RC, the analysis showed no clear trend, since the results are complex and hard to interpret.

mTOR mutation is likely to be non-functional or activating One of the questions I wanted to answer was whether the mutation in mTORC1 in cell line MZ1851RC is a knock-out or non-functional. In the specific pool Rp.1851 the negative feedback from mTORC1 on IGFR was enriched with approx. 93% of the models requiring this edge to be functional, even if we only consider models with a maximum of 5 optional edges the number is still high with 90%. I therefore conclude that the mutation is likely to not interrupt the negative feedback on IGFR.

On the contrary, the frequency of the feedback in Rp.1257 only reaches a level approx. 75% for the full set of optional edges and only about 55% for models with a maximum of 5 crosstalk edges. Since this cell line does not carry a mutation in mTORC1, the comparison of the frequency suggests that the mutation of mTOR in MZ1851RC could increase the activity towards IGFR.

Minimal models show required influences and counteract overfitting In order to increase the interpretability of the results of the classification in the exact analysis, I structured and reduced the table for the number of Sorafenib targets and only regarded the minimal models for each target (Tables 6.6 and 6.7). For both cell lines, there were models from no Sorafenib target to three targets present. Thus, I cannot make any statements about whether Sorafenib acts on IGFR, EGFR, Raf, or any combinations of those except that almost all options are possible in our results.

However, the analysis does allow to make structural comparisons between the cell lines, since there are differences for sparse models observable. Moreover, I focused on the minimal models to avoid the problem of overfitting, since the more edges a model has the easier it is to produce complex dynamics (for more details see Discussion in Chapter 8).

The minimal models give an overview about how the system could compensate the influence of the inhibitor to fit the data for different levels of influence. For this aspect the cell lines show similarities and differences in their model structures. First, for both cell lines the minimal number of required crosstalk edges stays the same independent on the number of Sorafenib targets. Although each target requires different crosstalk edges to be present, the total number remains the same, which is one edge for all targets in Rp.1257 and in Rp.1851 one crosstalk is required for the targets EGFR and Raf, and two for IGFR as Sora target.

For minimal models of MZ1257RC the crosstalk edges for multiple Sorafenib targets are the combinations of the single target models. For example, IGFR as only target requires either the feedback EGFR, PI3K) to be present and EGFR as single target also (EGFR, PI3K) or (Erk, mTORC1) see Table 6.6 **b**. Then the minimal models for Sorafenib targeting IGFR and EGFR are possible with either one of the three edges. This observation suggests that increasing the number of drug targets does not apply more pressure on the systems structure and that the structure of the system for combinatorial targets in this case can be predicted from the single target topologies.

For the cell line specific pool Rp.1851 this does not hold. The most striking conflict with this theory is shown in Table 6.7 **c**, where the combination of IGFR and EGFR as targets is absent in the pool although the single targets are present in **b**. Also, looking at models with all three targets affected, there are topologies lost from the single and dual target models. Thus, for this cell line we cannot find a general rule for combining the drug targets.

Crosstalk from MAPK on PI3K is often required, not vice versa Next, I wanted to take a closer look at the crosstalk edges in the minimal models of both cell lines. Excluding the models without Sorafenib targets, we find that none of the models contain the cross-activation from PI3K on Raf. This is surprising, since PI3K on Raf was found to be essential in the crosstalk study in first section and is well described by the literature.

From a more general perspective, the crosstalk edge (PI3K, Raf) is the only influence from the PI3K pathway on MAPK. On the other hand, many models contain a crosstalk from MAPK on PI3K through EGFR or Erk and only a few models are valid with active feedback only. For Rp.1257, these models all have IGFR as Sorafenib target and in Rp.1857 the feedback is present in most models. Thus, in the cell line MZ1257RC there seems to be a trend that either the PI3K pathway is directly

affected by Sora or, if the Sorafenib targets MAPK only, crosstalk from MAPK on PI3K is necessary. Again, for cell line MZ1851RC the results are not clear except that the cell line is strongly dependent on the feedback.

General questions and future studies One main issue of the analysis is the interpretation of the models without a Sorafenib target. Since the data clearly shows an effect of the drug on components in this pathway, I expected all models to have at least one target of Sorafenib to be influenced. In fact, the number of “no-target” models is relatively low in both cell lines. On one hand this is due to the fact that there are less combinations possible with only 5 out of 8 optional edges, but especially the pool Rp.1257 shows a very small number of models for this scenario. This result matches our expectations and the few models left could be artifacts of the formalism or biologically not feasible with the consequence of excluding them.

Other possible interpretations could be that the real target is in the model, but we are not looking at the right one or only partially, meaning that Sorafenib might act on one of the proposed targets plus another one which was not considered.

Another general question is, whether we assume Sorafenib to have the same targets in both cell lines. From a molecular biological view this should be the case, since the cell lines are very similar and therefore the mechanism should be the same. A possible exception would be that one receptor is not expressed in one cell line or a target is mutated in one cell line, however, this is not the case in this study. Even if we include the assumption that Sorafenib should have the same targets in both cell lines into our analysis, the results are still not expressive enough to make any statements about the effect of Sorafenib.

In order to overcome this lack of expressiveness, our collaborators are in the process of performing more experiments with these cell lines and Sorafenib. As a next step, the cell lines are stimulated with either EGF or IGF and then treated with the inhibitor (and vice versa) in order to dissect the effect of the MAPK pathway from the PI3K pathway. Moreover, this data should provide more insight in the regulatory mechanisms of EGFR and IGFR with Sorafenib and hopefully clarify which receptor is affected by the drug.

Unraveling the regulation of mTORC2

This chapter is a study on the uncertain regulation of mTORC2 by applying the workflow from Chapter 3. This work was published recently and passages of that paper were adopted here [97].

For this study, I identified 5 hypotheses from the literature for the regulation of mTORC2, where each was supported by a set of experiments. Here, I translated these hypotheses to uncertain edges on mTORC2 and analyzed the model pool for data from the original studies. Moreover, the formalism allows for easy implementation of *in silico* experiments, which in turn can be exploited for experimental design. I used the experimental design to further analyze the structure and properties of the model pool and finally, discussed the results in context of the original studies and further literature.

7.1 Biological background

The mammalian target of rapamycin (mTOR) is a highly conserved kinase across species, from yeast to humans, playing a central role in coordinating cell growth, metabolism and survival of the cell [56]. In the cell, mTOR acts as a signal integrator through two distinct complexes, mTORC1 and mTORC2, each phosphorylating distinct sets of substrates upon stimulation by growth factors, nutrients, hormones, stress, and other stimuli [116]. Dysregulation in these processes was found to be present in many cancer types, therefore understanding the structure and dynamics of mTOR regulation is of high interest [19]. Although mTORC1 was the main focus of most studies so far, recent studies found mTORC2 playing an important role in cancer development, e.g. in HER2/PIK3CA-hyperactive breast cancer [12]. The development of novel mTOR kinase inhibitors has already yielded interesting findings on mTORC1 and mTORC2, but in order to successfully apply these drugs in combined therapy, a detailed understanding of the signaling processes is essential and not yet achieved [26, 47].

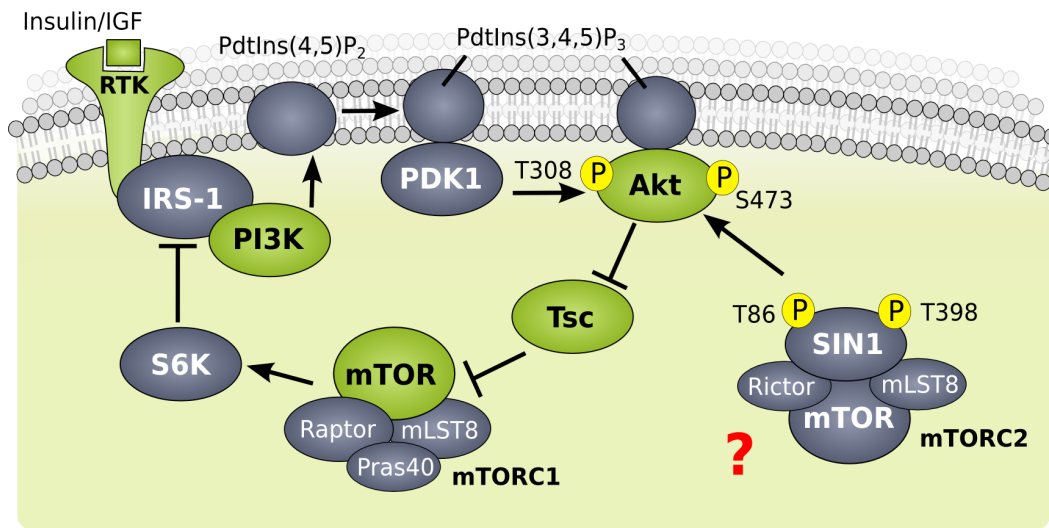


Fig. 7.1. Scheme of the PI3K pathway with candidate regulators of mTORC2 colored in green. Insulin and growth-factors activate RTK signaling through PI3K and the well-known regulation of mTORC1 with negative feedback on IRS-1. The regulation of mTORC2 is unclear.

Besides the catalytic mTOR subunit, both complexes contain mLST8, while Rictor and SIN1 are specific for mTORC2 and Raptor and Pras40 are specific for mTORC1. Upon stimulation with growth factors, mTORC1 is activated by PI3K signaling as described in Section 5.1 and mTORC2 activates AGC kinases (protein kinase A, G, and C), in particular it phosphorylates Akt at S473 [37]. Moreover, mTORC2 was reported to phosphorylate Akt at T450 at the mitochondrial membrane as a protein stabilizing post-translation modification. Since this process is growth-factor independent, it is not considered here.

7.1.1 Conflicting studies on mTORC2 regulation

In contrast to mTORC1, the processes that control mTORC2 are uncertain [32]. There are multiple studies investigating the influence of various kinases from the PI3K pathway on mTORC2 or components of its complex (see Fig. 7.1). Each of these studies was used as one hypothesis in our investigations: feedback independent regulation via RTK [22], activation by PI3K [30], positive feedback from Akt on mTORC2 [112], Tsc dependent regulation [43], and inhibition by mTORC1 [62].

Hypothesis 1: Feedback independent activation A feedback independent activation of mTORC2 was proposed by Dalle Pezze et al. [22], where they presented a data-driven ODE modeling approach investigating three different models: one model having Tsc as activator, a second model with only PI3K as activator and a third model where mTORC2 is regulated by an unknown kinase, which is independent from

the negative feedback on PI3K [22]. Here, the authors used experimental design based on simulations of the models to find perturbation experiments that are able to distinguish between the different hypotheses. Thereby, the group was able to extract their final model as a feedback independent activation of mTORC2.

Hypothesis 2: Direct activation by PI3K An activation of mTORC2 by PI3K was proposed by two different groups. Gan et al. [30] claimed that it is known that PI3K via PIP₃ has two effects on Akt. First, it recruits the kinase to the plasma membrane and phosphorylates the protein at T308. Secondly, it regulates the S473 phosphorylation of Akt via mTORC2, but whether or not it directly interacts with the complex is unknown [30]. Therefore, they created an Akt mutant which is constantly bound to the plasma membrane and thereby dissecting the recruiting effect from PIP₃ from its potential activation of mTORC2. Although they were able to show that the regulation via the Akt mutant is still sensitive to PI3K inhibitors, the exact mechanism could not be clarified [30].

In a recent work by Liu et al. (2015) a regulation of mTORC2 by PI3K was claimed, where they observed molecular interactions between SIN1, Akt and PIP₃ [61]. Liu et al. (2015) suggested that SIN1 might act as gate-keeper in mTORC2, therefore they investigated its mechanistic interaction with mTOR. As a result, the experiments showed that an interaction of SIN1-PH domain with the kinase domain of mTOR leads to a suppressed mTOR activity [61]. Since PH domains are characterized by their ability to bind PdtInsP_ns, Liu et al. (2015) tested binding properties of different PdtInsP_ns to SIN1-PH. They showed that PIP₃ binds to the SIN1-PH domain. Moreover, PIP₃ and SIN1 were shown to compete for binding with the kinetic domain of mTOR. Therefore Liu et al. (2015) claim that SIN1 binds mTORC2 blocking its activity and PIP₃ then binds SIN1 to release the inhibition on mTORC2, then Akt can bind to be phosphorylated.

Hypothesis 3: Akt directly activates mTORC2 causing a positive feedback Another member of the PI3K pathway, Akt, was proposed to regulate mTORC2 by two studies from the James lab [45, 112]. First, Humphrey et al. presented a quantitative analysis of the insulin signaling network in adipocytes using mass spectrometry-based proteomics [45]. In particular, they suggested that SIN1 phosphorylation at T86 is insulin sensitive and that this regulation acts through Akt, due to its timing and Akt inhibitor response. Moreover, a recent paper from the same lab by Yang et al. showed the same effect on a molecular level in various cell types [112]. Here, they examined SIN1 phosphorylation at T86 upon Akt, mTORC1 and S6K inhibition, showing a reduced phosphorylation level only for Akt inhibition but not mTORC1 or

S6K inhibition. They conclude that the activation of mTORC2 follows activation of Akt by T308 phosphorylation, then Akt phosphorylates SIN1 activating mTORC2, which itself then phosphorylates Akt at S473 for its full activation [112].

Hypothesis 4: Activation by Tsc2 Huang et al. [43] found that Tsc2, a component of Tsc, is required for mTORC2 activity by performing experiments with Tsc2 knock-out Mouse Embryonic Fibroblast (MEFs). For various stimuli they showed that in these cells the phosphorylation of Akt at S473 is lacking, but can be recovered adding a vector that expresses human Tsc2 [43]. Due to the negative feedback of mTORC1 on PI3K, a decreased activity of mTORC2 in Tsc2 knock-out cells can also result from constantly active mTORC1. In the paper, Huang et al. argue that the effect of the Tsc2 knock-out can be separated from the feedback by looking at experiments with mTORC1 inhibition.

Hypothesis 5: Integrity of mTORC2 is regulated by mTORC1 via SIN1 phosphorylation In direct contradiction with the findings of Humphrey and Yang et al., Liu et al. (2013) claimed in an earlier paper that S6K or Akt phosphorylates SIN1 not only at T86 but also at T398 and thereby causes a dissociation of the mTORC2 complex resulting in its inhibition [62]. In this paper, HeLa cells and MEF cells were stimulated with either insulin or EGF and treated with various inhibitors, mostly rapamycin but also S6K and Akt inhibitors. Moreover, SIN1 mutants with T96A and T398A genotype were used to mimic permanently non-phosphorylated SIN1 variants as well as knock outs.

7.2 Results

The results section follows the general workflow presented in Figure 3.1, where first the system is initialized, then the model pool is built, data sets are formalized, and the specific pool is analyzed using both the statistical and the exact analysis.

7.2.1 Model building from literature

For building a model of the mTORC2 regulation by signaling processes, I only included studies investigating direct interactions with the complex, excluding metabolic effects. I reduced the biological system to those components that are measured or perturbed in the studies we examined. The interactions between these components and their labels were also deduced from literature, where interactions that are widely accepted to be common knowledge were set to mandatory and uncertain

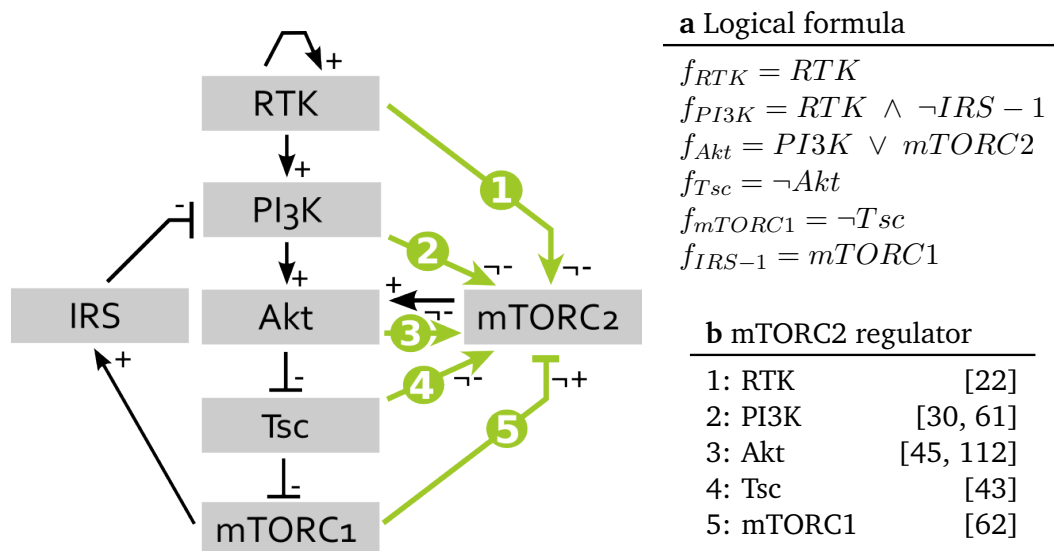


Fig. 7.2. Interaction graph and overview of hypothesis for mTORC2 regulation. Black lines indicate edges that are mandatory and green lines have edge labels allowing for uncertainty annotated with their respective edge label. **a** List of logical functions for components with known regulation, where the notation signifies a logical AND as \wedge , OR as \vee and negation as \neg . **b** List of candidate regulators for mTORC2 in the literature.

interactions as optional. Here, the regulations within the PI3K pathway and the negative feedback were assumed to be known. However, the regulation of mTORC2 is unclear, thus all candidate interactions from RTK, PI3K, Akt, Tsc and mTORC1 were set as optional.

The regulations of the components were defined as functions according to the edge labels. For components that only have one regulator, the function can be directly derived from the edge label. For PI3K and Akt, the logical connection between the regulators needed to be deduced from biological knowledge. PI3K is activated by RTK and inhibited by IRS-1, where IRS-1 binds and thereby blocks PI3K from interaction with other components including RTK. Thus, the logical connection is AND, since PI3K can only be active if RTK is active and IRS-1 is not. The interactions from PI3K and mTORC2 on AKT are connected with a logical OR, because both are able to activate the component independently [75].

The interaction graph of the model is shown in Figure 7.2, with the logical functions on the left side and a list of the regulators with references on the right. For the regulation of mTORC2 various hypotheses were published, which can be summarized as 5 candidate regulations:

1. RTK indirectly regulates mTORC2, thus the regulation is insensitive to the negative feedback of mTORC1 on PI3K [22]. For our model, we included an activating edge from RTK to mTORC2 with the label not inhibiting.
2. PI3K directly activates mTORC2 [30, 61], thus we included an activating edge from PI3K to mTORC2 labeled as not inhibiting in our model as Hypothesis 2.
3. Akt activates mTORC2 by phosphorylation of SIN1 at T86 [45, 112]. For our model, Hypothesis 3 is a not inhibiting edge from Akt to mTORC2.
4. Tsc is required for mTORC2 activity shown in Tsc2 knock-out cells [43], therefore we included this observation as Hypothesis 4 as a not inhibiting edge from Tsc to mTORC2.
5. A phosphorylation of SIN1 by S6K causes a disintegration of the complex [62], thus we included an inhibiting edge from mTORC1 on mTORC2 with the label not activating into our model.

In Figure 7.2, these regulations are marked as green lines, meaning that we do not know whether or not these connections are functional. Thus, the logical function for mTORC2 is uncertain and we wanted to explore all possible models using the edge labels and known logical rules for the other components as constraints. For building the model pool, every topology of possible combinations of the 5 candidate edges was created, resulting in 32 topologies. Then, for every topology all truth tables, representing a logical function, agreeing with the constraints were selected. This process is computationally challenging, e.g. for a component with n optional incoming edges the upper bound of possible truth tables is 2^{2^n} , which is then reduced by the considered constraints. Using Tremppi, a model pool of 7581 models was determined.

7.2.2 Data formalization

Subsequently, data from the original studies were used to filter the model pool for those models that are in agreement with the experimental data. For this aim, I discretized this data to match the logical formalism and encode it to make it accessible for our software. Here, we discretized data from each study which was included as a hypothesis for this investigation. Since the original studies have different levels of detail and used different methods to prove their hypothesis, I

can only include a subset of the performed experiments here (see Sec. 7.3 for more details).

Tab. 7.1. Redundancy in experiments across different studies.

	PI3K inh	mTORC1 inh	Tsc ko	insulin
Dalle Pezze et al.	Fig. 8 A	Fig. 7 A	Fig. 6 A,B	Fig. 4 A,B
Gan et al.	Fig. 2 A			
Liu et al. (2015)	Fig. 3 D			Fig. 2 D
Humphrey et al.				Fig. 6 B
Yang et al.	Fig. 4 C	Fig. 4 A,B		Fig. 4 B
Huang et al.		Fig. 3 A	Fig. 1 A	Fig. 3
Liu et al. (2013)		Fig. 1 A	Fig. S4 j	

The columns show types of experiments that were done in various studies yielding in matching qualitative behavior after discretization (data not shown).

Discretization of experiments from literature shows redundancy Even though the papers claim different results, I found that many performed the same or similar experiments from a qualitative perspective. For example, the time series measurements upon insulin stimulation were done by five out of seven studies. The resulting discretized sequences were partially redundant with data sets of other experiments (see Tab. 7.1). Similarly, an experiment with mTORC1 inhibition and insulin stimulation was done by four groups, either using rapamycin or shRNA against Raptor. After discretizing the data, all studies observed active PI3K and mTORC2 measured by Akt phosphorylation. The effect of PI3K inhibition on mTORC2 activation was studied by four groups, where inhibitors like Wortmannin or LY294002 were used to directly block PI3K or the activating connection to Akt was impeded by inhibiting PDK1. These experiments consistently led to inactive mTORC1 and mTORC2 across all studies. Three studies investigated the activity of mTORC2 upon insulin stimulation in Tsc knock out/down MEF cells with equivalent results.

For this study, I included the most comprehensive data set and excluded redundant information. These comparisons reduced the number of data sets to be tested to five different experiments shown in Table 2. Note that this observation indicates a certain reliability of the data, since even though the experiments were performed by different groups with different aims and setups, their qualitative interpretation is comparable.

Time-course measurements as well as knock down experiments from the study of Dalle Pezze et al. [22] were used. In detail, time series measurements of insulin stimulated HeLa cells for various proteins were done (see Fig. 4 in [22]). Here, I

included data of the following components for the filtering process: Akt-pT, Akt-pS, IRS-1-pS, and S6K-pT (see Tab. 7.2 **T_4B**). The data was discretized by mean value, then assigned to its designated readout. Here, RTK was added to the data set as a measured component to encode the stimulation over time. Finally, the sequence was encoded as CTL shown in Table 7.2 **T_4B**.

Also, data from the perturbation experiments from Dalle Pezze et al. was included, where mTORC1 was inhibited by shRNA against Raptor in HeLa cells and the phosphorylation levels of Akt were measured 45 and 100 minutes after insulin stimulation (see Fig. 7 in [22]). The corresponding CTL formula in Table 7.2 **T_7A** contains RTK as active in the initial state due to insulin stimulation and the knock down of Raptor is encoded as setting the logical equation of mTORC1 to 0 as a fixed component. Moreover, I assumed the signaling process to be in steady state, since there is no change even after 180 minutes observable.

The effect of PI3K inhibition on mTORC2 activation was studied by treating HeLa cells with different concentrations of the inhibitor Wortmannin, which directly blocks PI3K (see Fig. 8 in [22]). After stimulating the cells with insulin, inactive mTORC1 and mTORC2 was measured after 30 and 50 minutes, where the effect intensified with increasing concentration. The resulting CTL formula **T_8A** is shown in Table 1, where PI3K is fixed to zero due to the inhibition and the dynamics are assumed to be a fixpoint, since the behavior was stable over both time points.

Huang et al. used $Tsc2^{-/-}$ Mef cells and treated them with various stimuli for 30 minutes to measured Akt-pS as well as S6K-pT (see Fig. 1A in [43]). To encode the knock out, Tsc was fixed to zero and the stimulation encoded as active RTK. These experiments lead to active mTORC1 but inactive mTORC2 after e.g. insulin stimulation, resulting in the CTL formula **M_1A** (see Tab. 7.2). The authors expected this behavior to be stable over time, therefore I encoded this measurement as a fixpoint.

Also, Huang et al. investigated the influence of the negative feedback on the signaling process. In Fig. 3 B and C in [43], $Tsc2^{-/-}$ Mef cells were treated with insulin for 15 minutes and Akt-pS, IRS-1 and its binding to PI3K was measured. In the experiment, the phosphorylation of IRS-1 by mTORC1 was measured showing a hyperphosphorylation due to the knock-out in the mTORC1 inhibitor Tsc. In this hyperphosphorylated state of IRS-1 the binding with RTK and PI3K disintegrates and PI3K becomes inactive, thus IRS-1 is fixed to 1 in the CTL formula **M_3BC**. Additionally, they claimed that the impaired mTORC2 activity in $Tsc2^{-/-}$ Mef cells

Tab. 7.2. Data processing for logical analysis by discretization and formal encoding as CTL formula.

Property name: T_4B							CTL:	
measured	0	5	10	20	30	120	readout	EF (RTK=1&PI3K=1&mTORC2=1&
Akt-pT	0	1	0	1	1	0	PI3K	mTORC1=0&IRS-1=0&EF (RTK=1&PI3K=0&
Akt-pS	0	1	1	1	1	0	mTORC2	mTORC2=1&mTORC1=0&IRS-1=1&EF (RTK=1&
IRS-1-pS	0	0	1	0	1	1	IRS-1	PI3K=1&mTORC2=1&mTORC1=0&IRS-1=0&
p70S6K	0	0	0	1	1	1	mTORC1	EF (RTK=1&PI3K=1&mTORC2=1&
Insulin	1	1	1	1	1	1	RTK	mTORC1=1&IRS-1=1&EF (RTK=1&PI3K=0&
								mTORC2=0&mTORC1=1&IRS-1=1))))
								Initial State: RTK=1,
								PI3K=0,mTORC2=0,mTORC1=0,IRS-1=0
T_7A							CTL:	
measured	45	100	180	readout				EF(mTORC2=1 & PI3K=1 & Delta=0)
Akt-pT	1	1	1	PI3K				Initial State: RTK=1,
Akt-pS	1	1	1	mTORC2				Fixed: mTORC1=0
Insulin	1	1	1	RTK				
T_8A							CTL:	
measured	30	50	readout					EF(mTORC2=0 & mTORC1=0 & Delta=0)
Akt-pS	0	0	mTORC2					Initial State: RTK=1
p70-S6K	0	0	mTORC1					Fixed: PI3K=0
Insulin	1	1	RTK					
M_1A							CTL:	
measured	30	readout						EF(mTORC2=0 & mTORC1=1 & Delta=0)
Akt-pS	0	mTORC2						Initial State: RTK=1
S6K-pT	1	mTORC1						Fixed: Tsc=0
Insulin	1	RTK						
M_3BC							CTL:	
measured	15	readout						EF(mTORC2=0 & mTORC1=1 & Delta=0)
Akt-pS	0	mTORC2						Initial State: RTK=1
S6K-pT	1	mTORC1						Fixed: Tsc=0, IRS-1=1
Insulin	1	RTK						
M_3BC2							CTL:	
measured	15	readout						EF(mTORC2=0 & Delta=0)
Akt-pS	0	mTORC2						Initial State: RTK=1
S6K-pT	0	mTORC1						Fixed: Tsc=0, mTORC1=0
Insulin	1	RTK						

The tables show measured components, time points in minutes and readout. For the CTL formulas the settings are given, which is the measurements, the initial state and fixed components. If no measurement at time point 0 is available, the set up of the experiment is used, e.g. stimulation of the receptor. **T_4B** Time series data of selected components from Figure 4B in [22]. The table shows measurements that were discretized by mean value. CTL formula uses time point 0 as initial state and further data points as sequence. **T_7A** Perturbation experiment with knock down of mTORC1 component Raptor leads to sustained Akt activity, encoded as fixpoint in the CTL formula with fixed mTORC1 (Fig. 7 in [22]). **T_8A** PI3K inhibition by Wortmannin causes complete inhibition of all pathway components including Akt and mTORC1 target p70-S6K (Fig. 8 in [22]). The data is encoded as a fixpoint with fixed PI3K. **M_1A** Data from Huang et al., where Tsc2-/- cells show inactive mTORC1 and mTORC2, encoded as fixpoint with fixed Tsc (Fig. 1 in [43]). **M_3BC** and **M_3BC2** Combined data sets from two experiments for showing the independence of Tsc effect on mTORC2 and negative feedback (Fig. 3 in [43]), encoded as fixpoint with Tsc and IRS-1 fixed.

was not caused by a constantly activated feedback through mTORC1 on PI3K [43]. They argued that the mTORC2 activity should be rescued upon mTORC1 inhibition causing a deactivation of the feedback, if PI3K would be the only activator. To show this, mTORC1 was knocked down using siRNA against Raptor and the cells were stimulated with insulin for 15 minutes. In Fig. 3 B and C in [43], the binding of IRS-1 to PI3K was restored, but the mTORC2 complex did not show any kinetic activity. For the CTL formula **M_3BC2** in Table 7.2, Tsc and mTORC1 were fixed to zero. The data sets **M_3BC/2** were examined separated from the other experiments, because 15 minutes measurements are usually not sufficient to assume a fixpoint. However, we are especially interested in the effect of the feedback on the dynamics of the models.

Additionally to the experimental data, I assumed that without any stimulus the signaling system should reach an inactive steady state. This steady state represents the quiescence state of the biological system that is supposed to be fulfilled for the highly regulated growth-factor signaling in healthy tissue. Formally encoded, this means **Triv_FP**: $EF(mTORC2=0 \ \& \ \Delta t=0)$, **Initial State**: $RTK=0$.

Filtering for data reduced size of model pool Based on the data, we were able to fully determine the regulation for every component in the model, only the regulation of mTORC2 remains to be elucidated (Fig. 7.2). Combining all possible logical functions from 5 optional edges under the given constraints gives rise to 7,581 models, called initial pool. In the next step, this pool is filtered by applying CTL formulas derived from the data in Table 7.2 as restrictions on the model pool using model checking.

Tab. 7.3. Applying CTL formulas to the pool reduced its size markedly.

CTL:	Triv_Fp	T_4B	T_7A	T_8A	M_1A	M_3BC	M_3BC2	ExpD1	ExpD2
size:	5573	5202	7413	2008	7413	5573	168	2008	5573
is:	Red.pool: 944					944	0	310	634

Red.pool is the intersection (is) of all data sets except **M_3BC** and **M_3BC2**. **M_3BC** shows no further reduction on the Red.pool, whereas **M_3BC2** has no shared models with the Red.pool. On the right, the experimental design formulas are shown, which both show a further reduction on the selected pool.

As a result, each CTL formula reduces the initial pool to subpools of various sizes (see Tab. 7.3). Finally, the intersection of these subpools creates different reduced pools, which contains only those models that are valid for all CTL formulas. The main reduced pool is called Red.pool having 944 models that are true for all data sets excluding **M_3BC** and **M_3BC2**. Applying additional CTL formulas to the Red.pool,

the intersection with **M_3BC** did not result in a further reduction in the pool size meaning that all models in the Red.pool are valid for this data set. However, the opposite is true for **M_3BC2**, where no model from the Red.pool agrees with this formula. I will discuss this point further in the following section.

7.2.3 Model pool analysis

Although filtering the model pool for the data reduced its size markedly, 944 models are still too many to analyze them by hand. Therefore, we employ a statistical approach first, following up with an exact analysis.

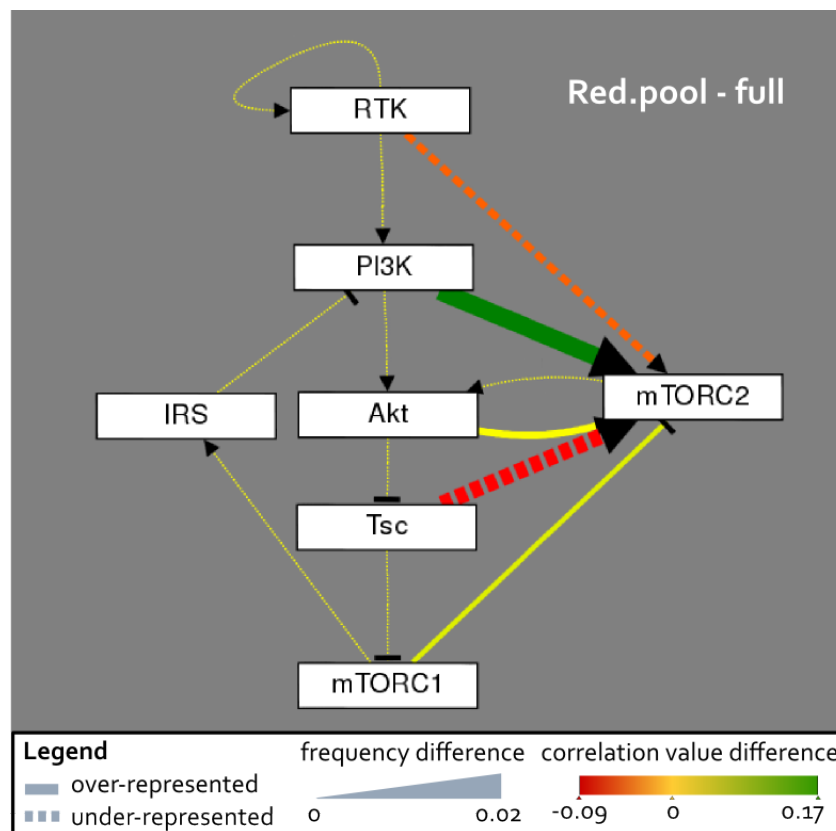


Fig. 7.3. PI3K regulation on mTORC2 is present in every filtered model, but not in original pool. Statistical analysis of the reduced pool and initial pool was created with Tremppi and the graph shows the difference (reduced - full) for the Red.pool. PI3K regulation of mTORC2 is overrepresented in the reduced pool in both frequency and impact compared to the initial pool. The regulation by RTK and Tsc is less frequent in the filtered pool than in the full shown by dashed lines, yellow dotted lines show identical frequency and impact in both pools.

Strong influence of PI3K regulation shown in statistical analysis In a first step, I evaluated both the reduced and the initial pool statistically using Tremppi. The difference was calculated by subtraction and visualized as a graph shown in Figure 7.3. The graph shows the difference between the Red.pool and the initial pool, where

an increase in frequency and impact for the regulation of mTORC2 by PI3K can be observed (Fig. 7.3). Moreover, the Red.pool contains less models with a regulation by RTK and Tsc than the initial pool and the impact of these edges is reduced. Note that the frequency and correlation differences are very small, since for frequency the maximum value is 0.02 in possible range of [0,1] and the correlation ranges between -0.09 and 0.17 with a possible range of [-2,2]. This means that the difference in frequency and impact between the pools is small, which can be explained as an artifact of the modeling process (more information in Sec. 7.3). In order to resolve these results further, we examine the composition of the pools explicitly.

Tab. 7.4. Exact analysis shows PI3K as essential regulator of mTORC2.

Edges	RTK	PI3K	Akt	Tsc	mTORC1	Size	
1	0	1	0	0	0	1	0.1%
2	1	1	0	0	0	1	0.1%
2	0	1	0	0	1	1	0.1%
3	0	1	1	0	1	3	0.3%
3	0	1	1	1	0	1	0.1%
3	1	1	0	0	1	1	0.1%
3	1	1	0	1	0	1	0.1%
3	1	1	1	0	0	1	0.1%
3	0	1	0	1	1	1	0.1%
3	0	1	1	0	1	1	0.1%
3	1	1	0	0	1	1	0.1%
4	0	1	1	1	1	10	1.1%
4	1	1	0	1	1	2	0.2%
4	1	1	1	0	1	24	2.5%
4	1	1	1	1	0	12	1.3%
4	0	1	1	1	1	10	1.1%
4	1	1	0	1	1	5	0.5%
4	1	1	1	0	1	5	0.5%
5	1	1	1	1	1	577	61.1%
5	1	1	1	1	1	286	30.3%

The table shows the classification of all 994 models in the Red.pool according to the following features: Edges in the model, the data sets, and active hypotheses. Size gives the number of models in the class and the percentage of this class in the pool.

Minimal model corresponds to Hypothesis 2 Despite the fact that the statistical evaluation is able to give us important information about the changes in the pool composition, it does not give information about explicit models. Thus, we performed an exact analysis in TomClass using as features: the number of edges, the validation for the CTL formulas, and the present hypotheses. Then the classification groups models that share the same topology and behavior towards the checked CTL formulas, and only differ in their logical equation.

Table 7.4 shows all 994 models in the Red.pool showing valid models with less than five edges. However, adding up the size of classes with five edges, it becomes clear that more than 90% of the models contain five edges as was expected, since the more regulators are available the easier the model can be fitted to the data. Moreover, all hypotheses are in agreement with the data, since every hypothesis is present in at least one model, even when only considering the models with less than five edges. In detail, three edges are necessary for all hypotheses to be present, for two edges models with pairwise combinations of {(mTORC1, mTORC2), (RTK, mTORC2), (PI3K, mTORC2)} are observed and only PI3K appears as possible single regulator. Thus, the minimal model, meaning the lowest numbers of mTORC2 regulators, corresponds to Hypothesis 2. Surprisingly, this edge is present in every model in the pool and therefore seems to be essential for the model dynamics to match the data. Thus, although all hypotheses are able to match the data, not all of them are necessary to be present.

Analysis of additional data set causes conflict I was especially interested in a data set by Huang et al., since they claimed to show an effect on mTORC2 that can be separated from the feedback affecting IRS-1 and PI3K [43]. Two CTL formulas, **M_3BC** and **M_3BC2**, were extracted from this data set and applied first as transient measurements. As a result, both formulas were in agreement with every model in the pool, since reaching one state is too easy for these very similar models in the pool (data not shown). Although the measurement time point was 15 minutes and therefore usually does not qualify for a steady state assumption, I tested the data as hypothetical fixpoints. Then, **M_3BC** was met by many models in the pool and the intersection with the Red.pool did not result in a further reduction of the pool (see Tab. 7.3).

However, the second formula **M_3BC2** led to a strong reduction in the pool size with only 169 out of 7581 being in agreement. When calculating the intersection with the Red.pool, the result is an empty set, caused by a direct conflict with **T_7A**. Therefore, our model does not support the conclusions drawn in the original paper (we will resolve this in more depth in the Sec. 7.3).

7.2.4 Experimental design

The idea of the experiment in Huang et al. to disrupt the feedback for dissecting the processes in the cascade and their effect on mTORC2 led me to propose a new experiment. For this experiment, I wanted to eliminate the negative feedback, e.g. by mutating the target phosphorylation side in IRS-1 such that IRS-1 maintains its

function as mediator of the signal from RTK to PI3K, but S6K cannot phosphorylate and inhibit PI3K. In such a system, a standard experiment would be to stimulate the receptor with insulin and measure the mTORC2 activity by AktpS levels.

From a modeling perspective, steady state measurements more effectively restrict the pool than transient measurements, therefore AktpS should be measured at multiple time points to ensure stability. The possible outcome of this experiment would be active or inactive mTORC2. To test this behavior on the Red.pool, I formulated these scenarios as CTL formulas:

ExpD1: $EF(mTORC2=0 \ \& \ \Delta=0)$, Initial State: $RTK=1$, Fix: $IRS-1=0$,

ExpD2: $EF(mTORC2=1 \ \& \ \Delta=0)$, Initial State: $RTK=1$, Fix: $IRS-1=0$.

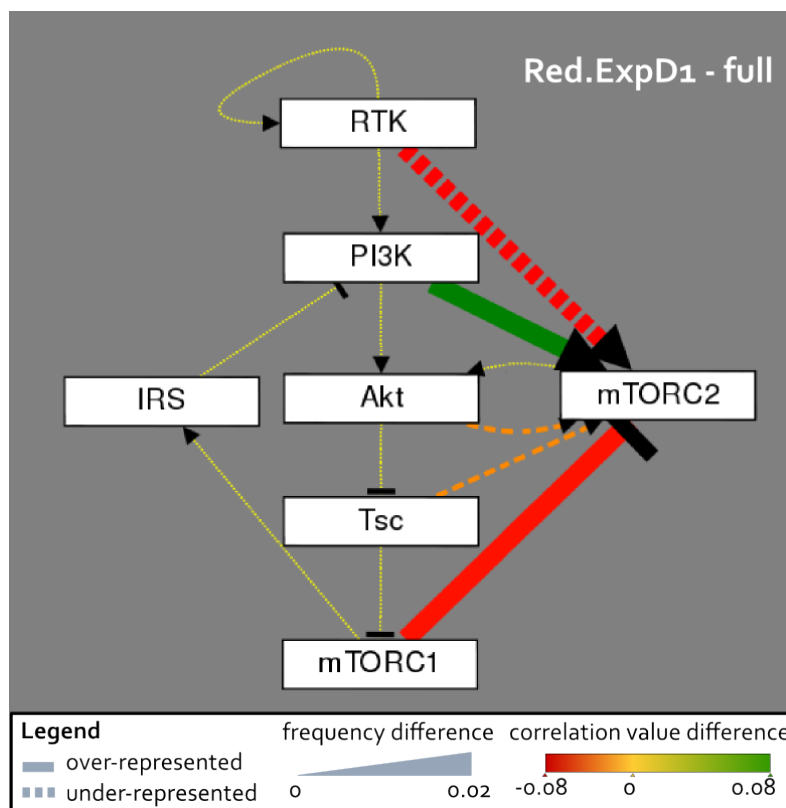


Fig. 7.4. Experimental design suggests mTORC1 as second regulator of mTORC2. The graph shows the difference of the statistical analysis of Red.ExpD1 and the initial pool, showing an over-representation of PI3K and mTORC1 as regulators and an under-representation of RTK.

Experiments split the pool for mTORC2 behavior The CTL formulas split the initial pool as well as the Red.pool in two groups, showing that every model reaches a fixpoint (Tab. 7.5). In both cases, the pool for **ExpD1** is roughly half the size of **ExpD2** with 310 to 634 models for the intersection with the Red.pool, called Red.ExpD1 and Red.ExpD2 respectively.

In order to further characterize the differences between these two pools, I analyzed both Red.ExpD1 and Red.ExpD2 analog to the Red.pool by a statistical and exact analysis. For Red.ExpD2, the results show no clear trend towards rejecting or supporting another hypothesis (Tab. 7.5). The minimal model with only the essential PI3K is in agreement with **ExpD2**, for two regulators only RTK is possible and for three regulators every hypothesis is present.

Tab. 7.5. Additional filtering for **ExpD1** and **ExpD2** yields two distinct pools.

	Edges	RTK	PI3K	Akt	Tsc	mTORC1	Size
Red.ExpD1	2	0	1	0	0	1	1
	3	0	1	0	1	1	1
	3	0	1	1	0	1	1
	3	1	1	0	0	1	1
	4	0	1	1	1	1	10
	4	1	1	0	1	1	5
	4	1	1	1	0	1	5
	5	1	1	1	1	1	286
Red.ExpD2	1	0	1	0	0	0	1
	2	1	1	0	0	0	1
	3	0	1	1	0	1	3
	3	0	1	1	1	0	1
	3	1	1	0	0	1	1
	3	1	1	0	1	0	1
	3	1	1	1	0	0	1
	4	0	1	1	1	1	10
	4	1	1	0	1	1	2
	4	1	1	1	0	1	24
	4	1	1	1	1	0	12
	5	1	1	1	1	1	577

The model pools Red.ExpD1 and Red.ExpD2 are listed with the same classification option than Table 7.4. **Red.ExpD1** shows the 310 models from the Red.pool that are in agreement with **ExpD1**, where all models contain PI3K and mTORC1 as essential regulators. In the second part, the 634 model agreeing with **ExpD2** and Red.pool do not show a clear tendency towards a second regulator.

In contrast, the analysis of Red.ExpD1 identified mTORC1 as second essential regulator, which is illustrated in the graph in Figure 7.4. Here, the difference between Red.ExpD1 and the initial pool shows an increase in frequency and impact for PI3K regulation, but it also displays a over-representation of mTORC1 inhibition of mTORC2. Furthermore, RTK is under-represented in the difference graph and has a negative impact (Fig. 7.4). In Red.ExpD1, the minimal model contains PI3K and mTORC1 as dual regulators for mTORC2, for three regulators every other hypothesis

is possible (Fig. 7.4). Thus, the data set **ExpD1** identifies a dual regulation of mTORC2 by PI3K and mTORC1 proposing an experiment to clarify this point.

7.3 Discussion

In this chapter, I applied the toolbox to investigate the uncertain regulation of mTORC2 by PI3K signaling. I was able to show that PI3K itself is necessary for mTORC2 activation, but the regulation is likely to be more complex. By enumerating all possible models arising from the state of the art literature, I systematically tested this pool of models for published data and analyzed the valid subpools. For analyzing these subpools, I first compared the reduced pools to the initial pool statistically. I was able to find enriched and under-represented hypotheses, but with a seemingly rather low significance.

The explanation for this issue is given by the exact analysis, where we observed that there is a bias towards models with many edges, i.e. more than 90% of models in the pool have five edges. There are two reasons for this bias: combinatorics and overfitting. When building the model pool, every possible logical expression is generated, where the number of combinations increases with the number of optional incoming edges. For a component with two regulators, the upper bound of possible truth tables is $2^{2^2} = 2^4$ and for five regulators it is $2^{2^5} = 2^{36}$. Also, the more regulators are allowed in a model the easier it is to produce complex dynamics, which is a common problem of overfitting.

More than 90% of the models of the initial and the specific pools have five edges, thus in the statistical analysis the difference for the frequency is only influenced by a maximum of 10% of the models. The impact also is biased by these models, since the impact automatically is split upon all regulators leading to a low impact in models with five edges. However, having four or even five kinases regulating one protein is unlikely. So even low values must be considered to uncover important trends. These are then validated in the second exact analysis to explicitly look at the minimal models of the pools.

In the exact analysis, models are grouped according to their number of optional edges. The analysis revealed that none of the hypotheses can be rejected, but require multiple edges to explain all data (see Fig. 7.3). On one hand this fact is surprising, because the original studies claimed different hypotheses for the mTORC2 regulation. On the other hand, the models are very similar and the data I used for filtering

the pool coincided with experiments from other studies after discretizing, thus discriminating between the models is hard.

Issue of selecting experimental data from studies I only used a subset of the performed experiments, which are not necessarily the most weighty arguments in the studies. Here, basic qualitative effects were examined, which cannot capture observations from experiments including specific manipulations of single components, such as mutating a phosphorylation site, or dose-dependent effects. For example, most data of Gan et al. contain an Akt mutant bound to the membrane which cannot be represented in our system without changing the model. Moreover, the level of detail varies among the studies, such that I selected a level of abstraction shared by all studies, e.g. I cannot include experiments with mutated SIN1, since represent a partial knock-down of mTORC2 cannot be represented.

The experimental setups between the studies differ, from cell types to treatments and methods. Most experiments used insulin as stimulus whereas the other studies used EGF. A very interesting data set from Liu et al. (2013) showed a transient deactivation of mTORC2 for EGF on a small time-scale (0 to 60 mins) (Fig. 3d in [62]), but for insulin this effect was only observable on a large-time scale (> 60 mins) (Fig. 3b in [62]). For such long time-scales it is questionable whether the observed effect is caused by signaling processes or might involve other processes outside the model boundaries. Also, EGF stimulation mainly activates the MAPK cascade, which is known to have crosstalk effects on PI3K signaling [107]. In order to maintain a minimum level of comparability, I did not include data with EGF stimulus. Nevertheless, the redundancy in the qualitative behavior in the experiments I observed in Table 7.1 affirms the comparability of the selected data sets.

Another data set I found to be interesting is from the study of Huang et al., which claimed to show a PI3K-independent effect on mTORC2 [43]. Here, the setup and the measured components fitted our model, resulting in two different CTL formulas, **M_3BC** and **M_3BC2** (Tab. 7.2), for two observations. However, this data set was a single measurement after 15 minutes of stimulation, thus it is not a steady state measurement. In general, many modeling formalisms require a steady state assumption. Although we are able to test both transient states and fixpoints, testing the reachability of one transient state is easy to fulfill by the models, thus every model was valid for the transient version. For this reason, I also tested both data sets also as hypothetical fixpoints of the system.

Comparison to original studies The analyses in Figure 7.3 and Table 7.4 revealed that PI3K is an essential regulator across all models in the filtered pools, which matches the results of Hypothesis 2 by Gan et al. even though I did not use any data from that study directly. Also, this finding supports a recent study from Yang et al. [112], where they suggested PIP₃ to act as a scaffold protein for the interaction between mTORC2 and Akt.

Comparing our results with the paper of Dalle Pezze et al., I sought for qualitative similarities and differences between the studies, since their investigations partially overlap with our studies. Since I included their final model as Hypothesis 1, I can say that our results do not rule out the existence of Hypothesis 1. In particular, the data sets from Dalle Pezze et al. **T_4B**, **T_7A** and **T_8A** do not exclude Hypothesis 1, thus there is no direct contradiction between the studies. Also their data sets have a large overlap with observations from other papers (see Tab. 7.1). However, I only used a subset of their data, since they measured the activity of mTORC2 by the phosphorylation of mTOR at S2481, for which there is a discussion on whether it is a unique read-out for the activity of the complex [18, 65, 21, 85]. Also, for fitting the models to the data, Dalle Pezze et al. added an unknown kinase which is able to phosphorylate Akt at S473 and thereby could substitute mTORC2.

Another interesting aspect is that Dalle Pezze et al. identified a PI3K variant as mTORC2 regulator, because it is sensitive to Wortmannin, but cannot be PI3K itself due to insensitivity to the negative feedback [22, 86]. This insensitivity was observed in a knock-down experiment for Raptor, where the phosphorylation of mTORC2 did not decrease upon feedback disruption. However, Raptor knock-down deactivates mTORC1, thus it also disrupts a potential inhibition by S6K on mTORC2. Therefore, our results propose an alternative solution for the PI3K variant by having PI3K and another second regulator. As a consequence, it would be very interesting to build an ODE model to be able to directly compare our models to Dalle Pezze et al. and to include quantitative information into the study.

Finally, the study of Huang et al. tested the behavior of mTORC2 in *Tsc2*^{-/-} Mef cells, where I selected three data sets. **M_1A** was the basic observation showing an impaired mTORC2 activity for the knock out, which was in agreement with almost all models in the initial pool (see Tab. 7.3). The data sets, **M_3BC** and **M_3BC2**, were not included in the Red.pool, but tested separately. The first formula, **M_3BC**, was in agreement with Red.pool showing that there is no more additional information contained in that data. The second formula, **M_3BC2**, which was the

main observation of Huang et al., resulted in no intersection with the Red.pool since it directly conflicts with the formula T_{7A} .

In the experiment, they tested whether an inhibition of mTORC1 can recover the activity of PI3K on mTORC2, since it deactivates the negative feedback. Huang et al. argue that this recovery was not observed, thus they conclude that PI3K cannot solely regulate mTORC2. For this purpose, a lot of perturbations were done, Tsc knock out, Raptor knock down, stimulation, but all these actions directly affect a possible regulator of mTORC2, therefore the expressiveness of the data is limited. To address this issue, I wanted to find an experiment that does not manipulate any of the hypothesized regulators of mTORC2, but gives more information about the feedback independent processes.

Identification of regulatory mechanisms requires deactivation of feedback There are two major reasons, why the exact regulation of mTORC2 by the PI3K pathway is hard to identify: (i) the candidates are within one signaling cascade and (ii) the negative feedback from mTORC1 on PI3K. From the first fact the problem of very short time windows arises, where a kinase becomes active without activating its downstream target, which is also a kinase. Producing data that is able to dissect the activity of kinases in a chain reaction is hard. A possible solution is to block the cascade at different levels using inhibitors as shown in Table 7.1, but due to the negative feedback this treatment affects all components in the pathway.

For this reason, I proposed an experiment, where the target phosphorylation of the negative feedback on IRS-1 are mutated. In detail, S302, S307 and S632 are causing a reduced signaling through PI3K when phosphorylated [36], therefore these serine residues would need to be substituted to e.g. alanine. When stimulating these mutants with insulin, I predicted two different outcomes for mTORC2, which splits the model pool in two groups for active and inactive mTORC2 in steady state. Analyzing the resulting model pools that agree with the data from the first analysis in the Red.pool, I found no clear pattern for Red.ExpD2 having active mTORC2 (Tab. 7.5). In contrast, Red.ExpD2 shows mTORC1 as a second essential regulator.

The edge from mTORC1 was reported by Liu et al. in form of a dual phosphorylation at Thr 86 and Thr 398 of the mTORC2 component SIN1. Whether or not these phosphorylations lead to mTORC2 inhibition [62] or for Thr 86 to activation by Akt as claimed by Humphrey et al. [45] is unclear, due to conflict of data [110]. Since Akt also regulates mTORC1, it remains to be clarified whether this effect is direct or indirect. Also, further studies on the exact mechanism through which modifications

of SIN1 affect the mTORC2 activity are necessary [110]. Still, the modifications of SIN1 suggest that a regulation by PI3K alone might not be realistic.

Moreover, in a recent summary Yuan et al. [115] propose a dual regulation of mTORC2, with PIP₃ as scaffold and recruiter as well as S6K as inhibitor, which matches our result from the experimental design pool Red.ExpD2. This finding is also supported by a recent study in C2C12 myoblasts using ODE modeling, where a regulation by PIP3 and S6K is proposed [7].

These considerations show that a potentially interesting next step could be to construct more detailed models in terms of modeling formalism or resolution of components, like mTORC2, to increase comparability and more fully exploit the available data. Such a study could lift our qualitative results to a more quantitative understanding of the mechanism of regulation of mTORC2.

Conclusion

Computational modeling methods are applied and designed for biological systems, but these methods only yield meaningful results if they are implemented and interpreted correctly. Here, I present a modular workflow for applying model checking tools that enable us to investigate pools of logical models [53, 90]. Within this approach, there are three steps that require a transfer of biological information into the mathematical formalism or vice versa. First, building the model and identifying a feasible objective requires understanding of the biological process and the formalism. I identified three different objectives that are biologically interesting in cancer signaling, and feasible in the formalism, which is crosstalk analysis, identifying driver mutations and testing the effect of drugs (see Fig. 3.1).

In a second step, the implementation of biological data is necessary for filtering out invalid models. Here, the encoding of the data with regard to temporal information, genotype, choice of strictness and monotonicity is required, which was shown to have a great impact on the results in the case studies. Finally, the third step is the analysis of the outcome of the filtering process which needs to be interpreted for new biological insight, where the toolbox employs a statistical and an exact analysis. Each step contains limitations and possible expansions, which are discussed in the following.

Technical limitations of the workflow Traditionally, logical models are used to model large systems, but here we are limited to small networks. Since our aim is to be able to cope with uncertainty in a system, the complexity in our approach originates from the size of the model pool. On one hand, the number of components is restricted, because the state space grows exponentially with the number of components and in order to test trajectories for model checking the full state space needs to be computed. Tremppi is able to cope with up to 35 components [90], TomClass is restricted to less [53]. On the other hand, the number of uncertain edges increases the number of models in the pool, which limits the analysis by run-time. Here, the increase in model numbers depends on the edge label and the possible parametrizations arising in combination with other edges.

In the case studies, I investigated biological systems with varying levels of complexity, where the number of components was relatively similar with 7 components in the mTORC2 study up to 14 in the EGFR study. However, the number of optional edges were very different among the case studies with a model pool of 259,200 models in the EGFR study with 13 optional edges (see Chapter 5), 19,404 in the Sorafenib study with 7 optional edges (see Chapter 6), and only 7,581 models in the mTORC2 study with 5 optional edges (see Chapter 7). Here, the Sorafenib and mTORC2 studies were feasible in TomClass, but the EGFR study required the optimized model checking tool Tremppi to perform the model checking in a reasonable time.

Issue of encoding data as temporal logics One difficulty using the model checking framework is the translation of experimental data into temporal logics. For checking CTL formulas, for example, the choice of verification type, i.e., whether the property needs to hold for all or only for some initial states, will impact the results. I tested this parameter in the crosstalk study, where Table 6.2 lists the different pool sizes for this option. In this example, there often was no difference between *ForAll* and *ForSome*, which is due to the model topology and the tested data. The interaction graph only has one input and most of the tested data was steady state data. Annotating the attractors to the models, I found that only a small fraction had more than two attractors, one for an active and one for an inactive receptor. In these models, all initial states sharing the same value for the receptor can reach the fixpoint if one initial state can reach it, since these models have two distinct basins of attraction for each attractor. However, this effect was not expected to be that clear, since there are optional feedbacks in the system which can cause multi-stationarity or cyclic attractors [100].

Finally, a transient data set in the study demonstrates the possible impact of the strictness parameter with a pool of more than 10,000 models for the relaxed and less than 1,000 models for the strict application. In general, both options are not optimal for the application to biological data. The relaxed option is too easy to be fulfilled especially considering that the asynchronous update creates a state space with many trajectories of which some might not be biological meaningful. Such a trajectory on the other hand might also be the reason why the strict option rejects a model, even if all but one trajectory are in agreement with the formula. Furthermore, biological data is noisy and the strict option has less tolerance towards errors than the relaxed option. For this reason, I opted for the low strictness option for all presented analysis. A helpful extension of the strictness parameter would be a percentage measure showing how many trajectories are valid. This measure could show how robust a model can simulate the dynamics, which is also interesting with regard to a possible

ranking of models within the pool. Tremppi has a measure called *cost* that shows how many transition were necessary to fulfill the formula, but it can only test the relaxed strictness parameter.

The second difficulty when encoding data into temporal logics is the decision on whether an observation is transient or stable. This impact is stressed in the EGFR study in Chapter 5, where Table 5.2 shows the pool sizes for different temporal interpretations of the same data. For signaling systems the decision on whether a measurement is at steady state or not is especially hard, since the time point should be late enough to be considered stable, but not too late since it is likely that non-signaling effects or influences from outside the model boundaries get involved. Often, biological knowledge allows for well-supported decisions in these matters, e.g. in the EGFR study we assumed experiments with stimuli as transient and without as stable (see Sec. 5.2.3).

Similar problems arise when discretizing data, where the interpretation of a measurement as active or inactive often is not clear. In this thesis, I presented two different kinds of data which require different processing. There is qualitative information such as Western Blots, where the intensity of a signal is interpreted relative a control, e.g. in Fig. 6.5. For quantitative data, a threshold needs to be defined, e.g. by using various methods [25] which can result in different thresholds. For this reason, I do not apply these measures in a strict way, but consider information on downstream components for the discretization process. In case the activity of a component is just below the threshold but its target is active, the component is set to be active. Another issue is data with very small variations where it is uncertain, whether this variation is a functional difference or noise. In the EGFR study, we used three levels and the fold-change as a measure to discriminate upregulation, downregulation and no change (see example in Tab. 5.1). In the Sorafenib study, the Bio-Plex data set contained components which showed very low intensities throughout the experiments. For example, the receptor VEGFR was interpreted as not expressed in the cells such that the component was deleted from the model (Fig. A.6).

Analysis of model pools and experimental design There are two different kinds of analyses used in this thesis: a statistical and an exact analysis. The statistical analysis in Tremppi has the advantage of identifying general trends by visualizing the frequency and impact of a pool and the option to compare these measures with the other pools. Thereby, the effect of the data through the filtering process can be identified. In the case studies, this analysis revealed limitations in terms of significance and interpretability. The low significance values, as discussed in

Chapter 7, originate from the problem of overfitting, which means that models with more edges more easily fit the data and make up a larger proportion in the pool. This results in a strong bias towards complex models. A possible improvement for the statistical analysis could be a change in the impact measure, which penalizes redundant influences, e.g. using mutual information [101]. Here, the information flow is split into synergistic, redundant and unique influences from all information sources. Mutual information would increase the significance of unique influences from logical OR connections and synergistic influences from logical AND connections, as well as lower the significance from redundant influences.

In general, the biological systems modeled in this thesis are expected to be sparse. In the signaling pathways, the components are single proteins or complexes, where an interaction often describes a molecular interaction for which the component is specialized. This means, I specifically look for models with a small number of interactions in the analysis of model pools. For this aim, the exact analysis using TomClass delivered meaningful results in all case studies. With the classification tool it is possible to list models according to features like optional edges and agreement with data. In contrast to the statistical analysis, we do not analyze the mean properties of the pool, but look at single models. Thereby, biologically relevant information can be identified such as edges or conflicting data sets. However, this analysis becomes infeasible for very large pools, thus very large classification tables. In the Sorafenib case study, the analysis was only manageable with a clearly formulated aim and a time intensive, error-prone processing of the table.

The analysis of model pools is mainly burdened by the redundancy that results from the enumeration of all possible models. As mentioned before, the statistical analysis could be improved by changing the correlation measure and the exact analysis could be eased by methods that systematically find characteristics in databases, e.g. Logical Analysis of Data (LAD). This is a classification method that could be applied to the database in order to learn so-called patterns of features that are true for an observation [5]. For example, minimal patterns in LAD would correspond to the minimal models in the thesis.

Future work and experimental design The toolbox presented in this thesis provides a guideline to model and assess uncertainty in logic models. However, there is room for improvements and extensions, especially considering the objectives and the pool analysis. As possible objective, experimental design on model pools would be an interesting aim. In the case study of mTORC2, I already used experimental design to improve the understanding of the final model pool by creating hypothetical datasets

that could be translated to an experiment. As an objective, experimental design could involve systematic perturbations of drugable components to classify the pool into “treatment groups” for experiments.

Moreover, the crosstalk between signaling pathways has become a major focus in cancer treatment [38, 23, 42, 70, 77, 79, 82, 87, 94]. Using the crosstalk analysis, I presented an approach to integrate existing models of single pathways into one model, while preserving their dynamical characteristics [98]. Hence, as much prior knowledge as possible is used from existing studies and models, to literature information and uncertain information or hypotheses. The resulting pool of models can also be tested for single drug treatment or combinations. For modeling cancer systems, a model pool might even be the more accurate description of a tumor than one single model, since tumors are heterogeneous. Thus by finding a treatment that affects a pool of models could be more beneficial than optimizing the treatment for a single model, which only represents a fraction of the tumor. Here, we could aid the development of drugs by testing which component or combination of components would need to be inhibited to observe a desired read out in as many models as possible in the pool.

In short, I see this toolbox as a helpful analysis approach for modeling small signaling or gene regulatory systems before applying more detailed and often error-prone modeling formalisms like ODE systems. Even though this approach requires a strong simplification of biological processes, the application of the toolbox can deliver meaningful results as shown in the mTORC2 case study in Chapter 7.

Bibliography

- [1] E. Aksamitiene, A. Kiyatkin, and B. N. Kholodenko. Cross-talk between mitogenic Ras/MAPK and survival PI3K/Akt pathways: a fine balance. *Biochemical Society Transactions*, 40(1):139–146, 2012.
- [2] I. Albert, J. Thakar, S. Li, R. Zhang, and R. Albert. Boolean network simulations for life scientists. *Source code for biology and medicine*, 3(1):1, 2008.
- [3] U. Alon. Biological networks: the tinkerer as an engineer. *Science*, 301(5641):1866–1867, 2003.
- [4] C. Baier, J.-P. Katoen, et al. *Principles of model checking*, volume 26202649. MIT press Cambridge, 2008.
- [5] K. Becker, M. Gebser, T. Schaub, and A. Bockmayr. Answer Set Programming for Logical Analysis of Data. In *Workshop on Constraint-Based Methods for Bioinformatics (WCB'16)*, page 15, 2016.
- [6] G. Bernot and J.-P. Comet. On the use of temporal formal logic to model gene regulatory networks. In *Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 112–138. Springer, 2009.
- [7] A. Bertuzzi, F. Conte, G. Mingrone, F. Papa, S. Salinari, and C. Sinisgalli. Insulin Signaling in Insulin Resistance States and Cancer: A Modeling Analysis. *PloS one*, 11(5):e0154415, 2016.
- [8] F. J. Bruggeman, H. V. Westerhoff, J. B. Hoek, and B. N. Kholodenko. Modular response analysis of cellular regulatory networks. *Journal of theoretical biology*, 218(4):507–520, 2002.
- [9] M. Burotto, V. L. Chiou, J.-M. Lee, and E. C. Kohn. The MAPK pathway across different malignancies: a new perspective. *Cancer*, 120(22):3446–3456, 2014.
- [10] A. Carriere, Y. Romeo, H. A. Acosta-Jaquez, J. Moreau, E. Bonneil, P. Thibault, D. C. Fingar, and P. P. Roux. ERK1/2 phosphorylate Raptor to promote Ras-

dependent activation of mTOR complex 1 (mTORC1). *Journal of Biological Chemistry*, 286(1):567–577, 2011.

- [11] M. Carrillo, P. A. Góngora, and D. A. Rosenblueth. An overview of existing modeling tools making use of model checking in the analysis of biochemical networks. *Front Plant Sci*, 3(155):1–13, 2012.
- [12] Y. Chen, J. Qian, Q. He, H. Zhao, L. Toral-Barza, C. Shi, X. Zhang, J. Wu, and K. Yu. mTOR complex-2 stimulates acetyl-CoA and de novo lipogenesis through ATP citrate lyase in HER2/PIK3CA-hyperactive breast cancer. *Oncotarget*, 7(18):25224–25240, 2016.
- [13] L. Chin, J. N. Andersen, and P. A. Futreal. Cancer genomics: from discovery science to personalized medicine. *Nature medicine*, 17(3):297–303, 2011.
- [14] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In *International Conference on Computer Aided Verification*, pages 359–364. Springer, 2002.
- [15] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri. Nusmv: a new symbolic model checker. *International Journal on Software Tools for Technology Transfer*, 2(4):410–425, 2000.
- [16] E. Clarke, K. McMillan, S. Campos, and V. Hartonas-Garmhausen. Symbolic model checking. In *International Conference on Computer Aided Verification*, pages 419–422. Springer, 1996.
- [17] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2):244–263, 1986.
- [18] J. Copp, G. Manning, and T. Hunter. TORC-specific phosphorylation of mammalian target of rapamycin (mTOR): phospho-Ser2481 is a marker for intact mTOR signaling complex 2. *Cancer research*, 69(5):1821–1827, 2009.
- [19] K. D. Courtney, R. B. Corcoran, and J. A. Engelman. The PI3K pathway as drug target in human cancer. *Journal of Clinical Oncology*, 28(6):1075–1083, 2010.
- [20] C. M. Croce. Causes and consequences of microRNA dysregulation in cancer. *Nature reviews genetics*, 10(10):704–714, 2009.
- [21] P. Dalle Pezze, A. G. Sonntag, D. P. Shanley, and K. Thedieck. Response to Comment on A Dynamic Network Model of mTOR Signaling Reveals TSC-Independent mTORC2 Regulation: Building a Model of the mTOR Signaling

Network with a Potentially Faulty Tool. *Science Signaling*, 5(232):lc4–lc4, 2012.

- [22] P. Dalle Pezze, A. G. Sonntag, A. Thien, M. T. Prentzell, M. Gödel, S. Fischer, E. Neumann-Haefelin, T. B. Huber, R. Baumeister, D. P. Shanley, et al. A dynamic network model of mTOR signaling reveals TSC-independent mTORC2 regulation. *Science signaling*, 5(217):ra25–ra25, 2012.
- [23] A. De Luca, M. R. Maiello, A. D'Alessio, M. Pergameno, and N. Normanno. The RAS/RAF/MEK/ERK and the PI3K/AKT signalling pathways: role in cancer pathogenesis and implications for therapeutic approaches. *Expert opinion on therapeutic targets*, 16(sup2):S17–S27, 2012.
- [24] R. De Smet and K. Marchal. Advantages and limitations of current network inference methods. *Nature Reviews Microbiology*, 8(10):717–729, 2010.
- [25] E. S. Dimitrova, M. P. V. Licon, J. McGee, and R. Laubenbacher. Discretization of time series data. *Journal of Computational Biology*, 17(6):853–868, 2010.
- [26] R. J. Dowling, I. Topisirovic, B. D. Fonseca, and N. Sonenberg. Dissecting the role of mTOR: lessons from mTOR inhibitors. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1804(3):433–439, 2010.
- [27] J. A. Engelman, J. Luo, and L. C. Cantley. The evolution of phosphatidylinositol 3-kinases as regulators of growth and metabolism. *Nature Reviews Genetics*, 7(8):606–619, 2006.
- [28] Z. Feng and A. J. Levine. The Regulation of Energy Metabolism and the IGF-1/mTOR pathways by the p53 Protein. *Trends in cell biology*, 20(7):427–434, 2010.
- [29] C. A. Gallo, R. L. Cecchini, J. A. Carballido, S. Micheletto, and I. Ponzoni. Discretization of gene expression data revised. *Briefings in bioinformatics*, page bbv074, 2015.
- [30] X. Gan, J. Wang, B. Su, and D. Wu. Evidence for direct activation of mTORC2 kinase activity by phosphatidylinositol 3,4,5-trisphosphate. *Journal of Biological Chemistry*, 286(13):10998–11002, 2011.
- [31] M. J. Garnett and R. Marais. Guilty as charged: B-RAF is a human oncogene. *Cancer cell*, 6(4):313–319, 2004.
- [32] C. Gaubitz, M. Prouteau, B. Kusmider, and R. Loewith. TORC2 Structure and Function. *Trends in biochemical sciences*, 41(6):532–545, 2016.

- [33] D. M. Graham, V. M. Coyle, R. D. Kennedy, and R. H. Wilson. Molecular Subtypes and Personalized Therapy in Metastatic Colorectal Cancer. *Current Colorectal Cancer Reports*, pages 1–10, 2016.
- [34] C. Greenman, P. Stephens, R. Smith, G. L. Dalgliesh, C. Hunter, G. Bignell, H. Davies, J. Teague, A. Butler, C. Stevens, et al. Patterns of somatic mutation in human cancer genomes. *Nature*, 446(7132):153–158, 2007.
- [35] L. Grieco, L. Calzone, I. Bernard-Pierrot, F. Radvanyi, B. Kahn-Perlès, and D. Thieffry. Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLoS computational biology*, 9(10):e1003286, 2013.
- [36] P. Gual, Y. Le Marchand-Brustel, and J.-F. Tanti. Positive and negative regulation of insulin signaling through IRS-1 phosphorylation. *Biochimie*, 87(1):99–109, 2005.
- [37] D. A. Guertin, D. M. Stevens, C. C. Thoreen, A. A. Burds, N. Y. Kalaany, J. Moffat, M. Brown, K. J. Fitzgerald, and D. M. Sabatini. Ablation in mice of the mTORC components raptor, rictor, or mLST8 reveals that mTORC2 is required for signaling to Akt-FOXO and PKC α , but not S6K1. *Developmental cell*, 11(6):859–871, 2006.
- [38] X. Guo and X.-F. Wang. Signaling cross-talk between TGF- β /BMP and other pathways. *Cell research*, 19(1):71–88, 2009.
- [39] M. A. Hamburg and F. S. Collins. The path to personalized medicine. *New England Journal of Medicine*, 363(4):301–304, 2010.
- [40] D. Hanahan and R. A. Weinberg. The hallmarks of cancer. *cell*, 100(1):57–70, 2000.
- [41] D. Hanahan and R. A. Weinberg. Hallmarks of cancer: the next generation. *cell*, 144(5):646–674, 2011.
- [42] H. Hu, A. Goltsov, J. L. Bown, A. H. Sims, S. P. Langdon, D. J. Harrison, and D. Faratian. Feedforward and feedback regulation of the MAPK and PI3K oscillatory circuit in breast cancer. *Cellular signalling*, 25(1):26–32, 2013.
- [43] J. Huang, C. C. Dibble, M. Matsuzaki, and B. D. Manning. The TSC1-TSC2 complex is required for proper activation of mTOR complex 2. *Molecular and cellular biology*, 28(12):4104–4115, 2008.
- [44] K. Huang and D. C. Fingar. Growing knowledge of the mTOR signaling network. In *Seminars in cell & developmental biology*, volume 36, pages 79–90. Elsevier, 2014.

- [45] S. J. Humphrey, G. Yang, P. Yang, D. J. Fazakerley, J. Stöckli, J. Y. Yang, and D. E. James. Dynamic adipocyte phosphoproteome reveals that Akt directly regulates mTORC2. *Cell metabolism*, 17(6):1009–1020, 2013.
- [46] T. Ideker and D. Lauffenburger. Building with a scaffold: emerging strategies for high-to low-level cellular modeling. *Trends in biotechnology*, 21(6):255–262, 2003.
- [47] E. Ilagan and B. D. Manning. Emerging Role of mTOR in the Response to Cancer Therapeutics. *Trends in Cancer*, 2(5):241–251, 2016.
- [48] S. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, 1969.
- [49] B. N. Kholodenko. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. *European Journal of Biochemistry*, 267(6):1583–1588, 2000.
- [50] B. N. Kholodenko. Cell-signalling dynamics in time and space. *Nature reviews Molecular cell biology*, 7(3):165–176, 2006.
- [51] H. Kitano. Biological robustness. *Nature Reviews Genetics*, 5(11):826–837, 2004.
- [52] S. Klamt, J. Saez-Rodriguez, and E. D. Gilles. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC systems biology*, 1(1):1, 2007.
- [53] H. Klarner. *Contributions to the Analysis of Qualitative Models of Regulatory Networks*. PhD thesis, Freie Universität Berlin, 2014.
- [54] B. Klinger, A. Sieber, R. Fritsche-Guenther, F. Witzel, L. Berry, D. Schumacher, Y. Yan, P. Durek, M. Merchant, R. Schäfer, et al. Network quantification of EGFR signaling unveils potential for targeted combination therapy. *Molecular systems biology*, 9(1), 2013.
- [55] C. Kuznia. *Molecular Mechanisms of Sorafenib-induced Apoptosis in Cancer Cells*. PhD thesis, Humboldt-Universität zu Berlin, 2015.
- [56] M. Laplante and D. M. Sabatini. mTOR signaling in growth control and disease. *Cell*, 149(2):274–293, 2012.
- [57] N. Le Novère. Quantitative and logic modelling of molecular and gene networks. *Nature Reviews Genetics*, 16(3):146–158, 2015.

- [58] S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. 1998.
- [59] W. A. Lim. Designing customized cell signalling circuits. *Nature reviews Molecular cell biology*, 11(6):393–403, 2010.
- [60] L. Liu, Y. Cao, C. Chen, X. Zhang, A. McNabola, D. Wilkie, S. Wilhelm, M. Lynch, and C. Carter. Sorafenib blocks the RAF/MEK/ERK pathway, inhibits tumor angiogenesis, and induces tumor cell apoptosis in hepatocellular carcinoma model PLC/PRF/5. *Cancer research*, 66(24):11851–11858, 2006.
- [61] P. Liu, W. Gan, Y. R. Chin, K. Ogura, J. Guo, J. Zhang, B. Wang, J. Blenis, L. C. Cantley, A. Toker, et al. PtdIns (3, 4, 5) P3-dependent activation of the mTORC2 kinase complex. *Cancer discovery*, 5(11):1194–1209, 2015.
- [62] P. Liu, W. Gan, H. Inuzuka, A. S. Lazorchak, D. Gao, O. Arojo, D. Liu, L. Wan, B. Zhai, Y. Yu, et al. Sin1 phosphorylation impairs mTORC2 complex integrity and inhibits downstream Akt signalling to suppress tumorigenesis. *Nature cell biology*, 15(11):1340–1350, 2013.
- [63] J. S. Logue and D. K. Morrison. Complexity in the signaling network: insights from the use of targeted inhibitors in cancer therapy. *Genes & development*, 26(7):641–650, 2012.
- [64] X. Long, Y. Lin, S. Ortiz-Vega, K. Yonezawa, and J. Avruch. Rheb binds and regulates the mTOR kinase. *Current Biology*, 15(8):702–713, 2005.
- [65] B. D. Manning. Comment on A dynamic network model of mTOR signaling reveals TSC-independent mTORC2 regulation: building a model of the mTOR signaling network with a potentially faulty tool. *Science signaling*, 5(232):lc3–lc3, 2012.
- [66] S. Marino, I. B. Hogue, C. J. Ray, and D. E. Kirschner. A methodology for performing global uncertainty and sensitivity analysis in systems biology. *Journal of theoretical biology*, 254(1):178–196, 2008.
- [67] S. Martin, Z. Zhang, A. Martino, and J.-L. Faulon. Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics*, 23(7):866–874, 2007.
- [68] L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla. Genetic control of flower morphogenesis in *Arabidopsis thaliana*: a logical analysis. *Bioinformatics*, 15(7):593–606, 1999.
- [69] M. C. Mendoza, E. E. Er, and J. Blenis. The Ras-ERK and PI3K-mTOR pathways: cross-talk and compensation. *Trends in Biochemical Sciences*, 36(6):320–328,

2011.

- [70] K. Moelling, K. Schad, M. Bosse, S. Zimmermann, and M. Schweneker. Regulation of Raf-Akt cross-talk. *Journal of Biological Chemistry*, 277(34):31099–31106, 2002.
- [71] P. T. Monteiro, D. Ropers, R. Mateescu, A. T. Freitas, and H. De Jong. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233, 2008.
- [72] A. Naldi, D. Berenguier, A. Fauré, F. Lopez, D. Thieffry, and C. Chaouiya. Logical modelling of regulatory networks with GINsim 2.3. *Biosystems*, 97(2):134–139, 2009.
- [73] J. D. Orth, I. Thiele, and B. Ø. Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.
- [74] V. W. Rebecca and K. S. Smalley. Change or die: targeting adaptive signaling to kinase inhibition in cancer cells. *Biochemical pharmacology*, 91(4):417–425, 2014.
- [75] V. S. Rodrik-Outmezguine, S. Chandarlapaty, N. C. Pagano, P. I. Poulikakos, M. Scaltriti, E. Moskatel, J. Baselga, S. Guichard, and N. Rosen. mTOR kinase inhibition causes feedback-dependent biphasic regulation of AKT signaling. *Cancer discovery*, 1(3):248–259, 2011.
- [76] P. P. Roux, B. A. Ballif, R. Anjum, S. P. Gygi, and J. Blenis. Tumor-promoting phorbol esters and activated Ras inactivate the tuberous sclerosis tumor suppressor complex via p90 ribosomal S6 kinase. *Proceedings of the National Academy of Sciences of the United States of America*, 101(37):13489–13494, 2004.
- [77] E. Rozengurt, H. P. Soares, and J. Sinnett-Smith. Suppression of Feedback Loops Mediated by PI3K/mTOR Induces Multiple Overactivation of Compensatory Pathways: An Unintended Consequence Leading to Drug Resistance. *Molecular cancer therapeutics*, 13(11):2477–2488, 2014.
- [78] J. Saez-Rodriguez, L. Simeoni, J. A. Lindquist, R. Hemenway, U. Bommhardt, B. Arndt, U.-U. Haus, R. Weismantel, E. D. Gilles, S. Klamt, et al. A logical model provides insights into T cell receptor signaling. *PLoS Comput Biol*, 3(8):e163, 2007.
- [79] K. S. Saini, S. Loi, E. de Azambuja, O. Metzger-Filho, M. L. Saini, M. Ignatiadis, J. E. Dancey, and M. J. Piccart-Gebhart. Targeting the PI3K/AKT/mTOR and Raf/MEK/ERK pathways in the treatment of breast cancer. *Cancer treatment reviews*, 39(8):935–946, 2013.

- [80] R. Samaga, J. Saez-Rodriguez, L. G. Alexopoulos, P. K. Sorger, and S. Klamt. The logic of EGFR/ErbB signaling: theoretical properties and analysis of high-throughput data. *PLoS computational biology*, 5(8):e1000438, 2009.
- [81] J. Sandoval and M. Esteller. Cancer epigenomics: beyond genomics. *Current opinion in genetics & development*, 22(1):50–55, 2012.
- [82] R. J. Shaw and L. C. Cantley. Ras, PI(3)K and mTOR signalling controls tumour cell growth. *Nature*, 441(7092):424–430, 2006.
- [83] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.
- [84] E. Simao, E. Remy, D. Thieffry, and C. Chaouiya. Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in *E. coli*. *Bioinformatics*, 21(suppl 2):ii190–ii196, 2005.
- [85] G. A. Soliman, H. A. Acosta-Jaquez, E. A. Dunlop, B. Ekim, N. E. Maj, A. R. Tee, and D. C. Fingar. mTOR Ser-2481 autophosphorylation monitors mTORC-specific catalytic activity and clarifies rapamycin mechanism of action. *Journal of Biological Chemistry*, 285(11):7866–7879, 2010.
- [86] A. G. Sonntag, P. Dalle Pezze, D. P. Shanley, and K. Thedieck. A modelling–experimental approach reveals insulin receptor substrate (IRS)-dependent regulation of adenosine monophosphate-dependent kinase (AMPK) by insulin. *FEBS Journal*, 279(18):3314–3328, 2012.
- [87] M. L. Sos, S. Fischer, R. Ullrich, M. Peifer, J. M. Heuckmann, M. Koker, S. Heynck, I. Stückrath, J. Weiss, F. Fischer, et al. Identifying genotype-dependent efficacy of single and combined PI3K-and MAPK-pathway inhibition in cancer. *Proceedings of the National Academy of Sciences*, 106(43):18351–18356, 2009.
- [88] J. Stelling, P. Mendes, F. Tonin, E. Klipp, R. Zecchina, M. Heinemann, N. Przulj, J. Wodke, S. Stoma, H. Kaltenbach, et al. Defining modeling strategies for Systems Biology. In *Technical report, FutureSysBio Workshop*, 2011.
- [89] M. R. Stratton, P. J. Campbell, and P. A. Futreal. The cancer genome. *Nature*, 458(7239):719–724, 2009.
- [90] A. Streck. *Toolkit for Reverse Engineering of Molecular Pathways via Parameter Identification*. PhD thesis, Freie Universität Berlin, 2015.
- [91] A. Streck and H. Siebert. Tremppi 1.0.0. <http://dibimath.github.io/TREMPPI/>, 2015.

- [92] A. Streck, K. Thobe, and H. Siebert. Analysing cell line specific EGFR signalling via optimized automata based model checking. In *Computational Methods in Systems Biology*, pages 264–276. Springer, 2015.
- [93] A. Streck, K. Thobe, and H. Siebert. Data-driven optimizations for model checking of multi-valued regulatory networks. *Biosystems*, 149:125–138, 2016.
- [94] C. Sun and R. Bernards. Feedback and redundancy in receptor tyrosine kinase signaling: relevance to cancer therapies. *Trends in biochemical sciences*, 39(10):465–474, 2014.
- [95] J.-F. Tanti and J. Jager. Cellular mechanisms of insulin resistance: role of stress-regulated serine kinases and insulin receptor substrates (IRS) serine phosphorylation. *Current opinion in pharmacology*, 9(6):753–762, 2009.
- [96] C. Terfve, T. Cokelaer, D. Henriques, A. MacNamara, E. Goncalves, M. K. Morris, M. van Iersel, D. A. Lauffenburger, and J. Saez-Rodriguez. CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC systems biology*, 6(1):1, 2012.
- [97] K. Thobe, C. Sers, and H. Siebert. Unraveling the regulation of mTORC2 using logical modeling. *Cell Communication and Signaling*, 15(1):6–21, 2017.
- [98] K. Thobe, A. Streck, H. Klarner, and H. Siebert. Model integration and crosstalk analysis of logical regulatory networks. In *Computational Methods in Systems Biology*, pages 32–44. Springer, 2014.
- [99] R. Thomas. Regulatory networks seen as asynchronous automata: a logical description. *Journal of Theoretical Biology*, 153(1):1–23, 1991.
- [100] R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviour of biological regulatory networks—I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of mathematical biology*, 57(2):247–276, 1995.
- [101] N. Timme, W. Alford, B. Flecker, and J. M. Beggs. Synergy, redundancy, and multivariate information measures: an experimentalist’s perspective. *Journal of computational neuroscience*, 36(2):119–140, 2014.
- [102] S. Videla, C. Guziolowski, F. Eduati, S. Thiele, M. Gebser, J. Nicolas, J. Saez-Rodriguez, T. Schaub, and A. Siegel. Learning Boolean logic models of signaling networks with ASP. *Theoretical Computer Science*, 599:79–101, 2015.

- [103] U. Vijapurkar, L. Robillard, S. Zhou, M. Degtyarev, K. Lin, T. Truong, J. Tremayne, L. B. Ross, Z. Pei, L. S. Friedman, et al. mTOR kinase inhibitor potentiates apoptosis of PI3K and MEK inhibitors in diagnostically defined subpopulations. *Cancer letters*, 326(2):168–175, 2012.
- [104] K. H. Vousden and X. Lu. Live or let die: the cell's response to p53. *Nature Reviews Cancer*, 2(8):594–604, 2002.
- [105] R.-S. Wang, A. Saadatpour, and R. Albert. Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5):055001, 2012.
- [106] S. M. Wilhelm, C. Carter, L. Tang, D. Wilkie, A. McNabola, H. Rong, C. Chen, X. Zhang, P. Vincent, M. McHugh, et al. BAY 43-9006 exhibits broad spectrum oral antitumor activity and targets the RAF/MEK/ERK pathway and receptor tyrosine kinases involved in tumor progression and angiogenesis. *Cancer research*, 64(19):7099–7109, 2004.
- [107] M. Will, A. C. R. Qin, W. Toy, Z. Yao, V. Rodrik-Outmezguine, C. Schneider, X. Huang, P. Monian, X. Jiang, E. De Stanchina, et al. Rapid induction of apoptosis by PI3K inhibitors is dependent upon their transient inhibition of RAS-ERK signaling. *Cancer discovery*, pages CD–13, 2014.
- [108] J. N. Winter, L. S. Jefferson, and S. R. Kimball. ERK and Akt signaling pathways function through parallel mechanisms to promote mTORC1 signaling. *American Journal of Physiology - Cell Physiology*, 300(5):C1172–C1180, 2011.
- [109] K.-K. Wong, J. A. Engelman, and L. C. Cantley. Targeting the PI3K signaling pathway in cancer. *Current opinion in genetics & development*, 20(1):87–90, 2010.
- [110] J. Xie and C. G. Proud. Signaling crosstalk between the mTOR complexes. *Translation*, 2(1):e28174, 2014.
- [111] N. Yaktapour, R. Übelhart, J. Schüler, K. Aumann, C. Dierks, M. Burger, D. Pfeifer, H. Jumaa, H. Veelken, T. Brummer, et al. Insulin-like growth factor-1 receptor (IGF1R) as a novel target in chronic lymphocytic leukemia. *Blood*, 122(9):1621–1633, 2013.
- [112] G. Yang, D. S. Murashige, S. J. Humphrey, and D. E. James. A positive feedback loop between Akt and mTORC2 via SIN1 phosphorylation. *Cell reports*, 12(6):937–943, 2015.
- [113] S.-H. Yang, A. D. Sharrocks, and A. J. Whitmarsh. MAP kinase signalling cascades and transcriptional regulation. *Gene*, 513(1):1–13, 2013.

- [114] C. F. Yu, Z.-X. Liu, and L. G. Cantley. ERK negatively regulates the epidermal growth factor-mediated interaction of Gab1 and the phosphatidylinositol 3-kinase. *Journal of Biological Chemistry*, 277(22):19382–19388, 2002.
- [115] H.-X. Yuan and K.-L. Guan. The SIN1-PH Domain Connects mTORC2 to PI3K. *Cancer discovery*, 5(11):1127–1129, 2015.
- [116] R. Zoncu, A. Efeyan, and D. M. Sabatini. mtor: from growth signal integration to cancer, diabetes and ageing. *Nature reviews Molecular cell biology*, 12(1):21–35, 2011.

Supplementary data and figures

HCT116								
stimulator	inhibitor	type	Akt	Erk	GSK3	IRS-1	Mek	p70-S6K
0,1% BSA	DMSO	c	579	2035	396	752	1789	1464
0,1% BSA	DMSO	c	368	2207	304	627	1752	1398
0,1% BSA	Medium	c	485	1969	296	613	1632	1604
0,1% BSA	Medium	c	677	2261	382	458	1533	1449
IGF 1	DMSO	t	24588	3185	1843	762	1744	8058
0,1% BSA	AZD 6244	t	446	229	251	428	5893	516
0,1% BSA	AZD 6244	t	469	197	259	417	5877	499
IGF 1	AZD 6244	t	24306	212	1860	570	6547	6071
TGF_	AZD 6244	t	8054	1119	862	965	10049	3761
0,1% BSA	LY294002	t	256	3788	428	969	2329	2157
0,1% BSA	LY294002	t	206	3237	441	792	2113	1890
IGF 1	LY294002	t	2053	3645	615	859	2374	2312
0,1% BSA	DMSO	c	527	2118	345	613	1677	1479
IGF 1	DMSO	t	24588	3185	1843	762	1744	8058
0,1% BSA	DMSO	c	527	2118	345	613	1677	1479
TGF_	DMSO	t	446	229	251	428	5893	516
0,1% BSA	DMSO	c	527	2118	345	613	1677	1479
0,1% BSA	AZD 6244	t	458	213	255	423	5885	508
0,1% BSA	DMSO	c	527	2118	345	613	1677	1479
0,1% BSA	DMSO	c	527	2118	345	613	1677	1479
0,1% BSA	LY294002	t	231	3513	435	881	2221	2024
0,1% BSA	DMSO	c	527	2118	345	613	1677	1479
TGF_	LY294002	t	0	0	0	0	0	0
0,1% BSA	DMSO	c	0	-1	0	-1	-1	0
IGF 1	DMSO	t	1	-1	1	-1	-1	1
0,1% BSA	DMSO	c	-1	1	-1	-1	0	1
TGF_	DMSO	t	-1	0	-1	-1	1	0
0,1% BSA	DMSO	c	-1	1	-1	-1	0	1
0,1% BSA	AZD 6244	t	-1	0	-1	-1	1	0
0,1% BSA	DMSO	c	-1	-1	-1	-1	-1	-1
IGF 1	AZD 6244	t	-1	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	1	-1	-1	-1	-1	-1
0,1% BSA	LY294002	t	0	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	1	1	1	1	1	1
TGF_	LY294002	t	0	0	0	0	0	0
0,1% BSA	DMSO	c	0	-1	0	-1	-1	0
IGF 1	DMSO	t	1	-1	1	-1	-1	1
0,1% BSA	DMSO	c	0	-1	-1	-1	-1	0
TGF_	DMSO	t	1	-1	-1	-1	-1	1
0,1% BSA	DMSO	c	-1	1	-1	-1	0	1
0,1% BSA	AZD 6244	t	-1	0	-1	-1	1	0
0,1% BSA	DMSO	c	0	1	0	-1	0	0
IGF 1	AZD 6244	t	1	0	1	-1	1	1
0,1% BSA	DMSO	c	0	0	0	0	-1	0
TGF_	AZD 6244	t	1	1	1	1	-1	1
0,1% BSA	DMSO	c	1	-1	-1	-1	-1	-1
0,1% BSA	LY294002	t	0	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	-1	-1	-1	-1	-1
IGF 1	LY294002	t	1	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	1	0	0	0	-1	0
TGF_	LY294002	t	0	1	1	1	-1	1

Fig. A.1. Perturbation data set from Klinger et al. [54] for cell line HCT116. A detailed description of the experiment, see Figure A.2. Rows marked in grey are data sets that caused a conflict and were removed.

LIM1215								
stimulator	inhibitor	type	Akt	Erk	GSK3	IRS-1	Mek	p70-S6K
0,1% BSA	DMSO	c	142	361	201	97	1008	366
0,1% BSA	DMSO	c	145	361	128	89	800	321
0,1% BSA	Medium	c	139	375	244	78	1006	362
0,1% BSA	Medium	c	144	422	250	81	1154	354
IGF 1	DMSO	t	7989	312	747	105	671	1277
0,1% BSA	AZD 6244	t	235	155	235	96	764	314
0,1% BSA	AZD 6244	t	561	148	289	93	814	331
IGF 1	AZD 6244	t	6313	121	593	95	347	874
TGF_	AZD 6244	t	12811	814	1013	93	6009	2104
0,1% BSA	LY294002	t	179	456	540	91	1284	366
0,1% BSA	LY294002	t	157	580	401	97	1440	380
IGF 1	LY294002	t	298	481	200	80	741	330
TGF_	LY294002	t	163	2863	1048	96	3596	1829
0,1% BSA	DMSO	c	143	380	206	86	992	351
IGF 1	DMSO	t	7989	312	747	105	671	1277
0,1% BSA	DMSO	c	143	380	206	86	992	351
TGF_	DMSO	t	2793	2806	983	104	3416	2708
0,1% BSA	DMSO	c	143	380	206	86	992	351
0,1% BSA	AZD 6244	t	398	152	262	95	789	323
0,1% BSA	DMSO	c	143	380	206	86	992	351
IGF 1	AZD 6244	t	6313	121	593	95	347	874
0,1% BSA	DMSO	c	143	380	206	86	992	351
0,1% BSA	LY294002	t	168	518	471	94	1362	373
0,1% BSA	DMSO	c	143	380	206	86	992	351
TGF_	LY294002	t	163	2863	1048	96	3596	1829
0,1% BSA	DMSO	c	0	-1	0	-1	-1	0
IGF 1	DMSO	t	1	-1	1	-1	-1	1
0,1% BSA	DMSO	c	0	0	0	-1	0	0
TGF_	DMSO	t	1	1	1	-1	1	1
0,1% BSA	DMSO	c	0	1	-1	-1	-1	-1
0,1% BSA	AZD 6244	t	1	0	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	1	0	-1	1	0
IGF 1	AZD 6244	t	1	0	1	-1	0	1
0,1% BSA	DMSO	c	0	0	0	-1	0	0
TGF_	AZD 6244	t	1	1	1	-1	1	1
0,1% BSA	DMSO	c	0	-1	0	-1	-1	0
IGF 1	DMSO	t	1	-1	1	-1	-1	1
0,1% BSA	DMSO	c	0	0	0	-1	0	0
TGF_	DMSO	t	1	1	1	-1	1	1
0,1% BSA	DMSO	c	0	1	-1	-1	-1	-1
0,1% BSA	AZD 6244	t	1	0	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	1	0	-1	1	0
IGF 1	AZD 6244	t	1	0	1	-1	0	1
0,1% BSA	DMSO	c	0	0	0	-1	0	0
TGF_	AZD 6244	t	1	1	1	-1	1	1
0,1% BSA	DMSO	c	-1	-1	0	-1	-1	-1
0,1% BSA	LY294002	t	-1	-1	1	-1	-1	-1
0,1% BSA	DMSO	c	0	-1	-1	-1	-1	-1
IGF 1	LY294002	t	1	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	-1	0	0	-1	0	0
TGF_	LY294002	t	-1	1	1	-1	1	1

Fig. A.2. Perturbation data set from Klinger et al. [54] for cell line LIM1215. The cell were treated with a stimulator, where BSA means no stimulus, and an inhibitor, where DMSO means no inhibition. The column type signifies the control measurements with a 'c' and the treated measurements with a 't'. According to Klinger et al. the Bio-Plex Protein Array System was used to measure P-AKT (S473), P-ERK1/2 (Thr202/Tyr204/Thr185/Tyr187), P-ERK2 (Thr185/Tyr187), P-IRS1 (S636/S639), P-MEK1 (S217/S221), and P-p70S6K(Thr421/S424) and quantified signals using Odyssey software [54]. Rows marked in grey are data sets that caused a conflict and were removed.

SW403								
stimulator	inhibitor	type	Akt	Erk	GSK3	IRS-1	Mek	p70-S6K
0,1% BSA	DMSO	c	1313	416	237	111	160	233
0,1% BSA	DMSO	c	2427	279	253	108	145	190
0,1% BSA	Medium	c	2100	395	260	127	162	242
0,1% BSA	Medium	c	1667	290	273	107	133	201
IGF 1	DMSO	t	23963	1201	1088	158	214	2359
TGF_	DMSO	t	5582	1805	494	150	362	1371
0,1% BSA	BMS 34554	t	1017	1047	229	99	200	458
0,1% BSA	AZD 6244	t	1884	148	283	127	175	187
0,1% BSA	AZD 6244	t	3208	122	294	121	137	181
IGF 1	AZD 6244	t	24683	121	1362	135	432	940
TGF_	AZD 6244	t	9452	147	658	150	1123	469
0,1% BSA	LY294002	t	271	622	162	111	180	356
0,1% BSA	LY294002	t	160	635	112	124	210	416
IGF 1	LY294002	t	8177	788	620	118	197	608
TGF_	SB216763	t	2004	1991	345	141	328	1342
0,1% BSA	DMSO	c	1877	345	256	113	150	217
IGF 1	DMSO	t	23963	1201	1088	158	214	2359
0,1% BSA	DMSO	c	1877	345	256	113	150	217
TGF_	DMSO	t	5582	1805	494	150	362	1371
0,1% BSA	DMSO	c	1877	345	256	113	150	217
0,1% BSA	AZD 6244	t	2546	135	289	124	156	184
0,1% BSA	DMSO	c	1877	345	256	113	150	217
IGF 1	AZD 6244	t	24683	121	1362	135	432	940
0,1% BSA	DMSO	c	1877	345	256	113	150	217
0,1% BSA	LY294002	t	216	629	137	118	195	386
0,1% BSA	DMSO	c	1877	345	256	113	150	217
TGF_	LY294002	t	2004	1991	345	141	328	1342
0,1% BSA	DMSO	c	0	0	0	-1	-1	0
IGF 1	DMSO	t	1	1	1	-1	-1	1
0,1% BSA	DMSO	c	0	0	-1	-1	0	0
TGF_	DMSO	t	1	1	-1	-1	1	1
0,1% BSA	DMSO	c	-1	1	-1	-1	-1	-1
0,1% BSA	AZD 6244	t	-1	0	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	1	0	-1	0	0
IGF 1	AZD 6244	t	1	0	1	-1	1	1
0,1% BSA	DMSO	c	1	-1	-1	-1	-1	-1
0,1% BSA	LY294002	t	0	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	-1	0	-1	-1	0	0
TGF_	LY294002	t	-1	1	-1	-1	1	1
0,1% BSA	DMSO	c	0	0	0	-1	-1	0
IGF 1	DMSO	t	1	1	1	-1	-1	1
0,1% BSA	DMSO	c	0	0	-1	-1	0	0
TGF_	DMSO	t	1	1	-1	-1	1	1
0,1% BSA	DMSO	c	-1	1	-1	-1	-1	-1
0,1% BSA	AZD 6244	t	-1	0	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	1	0	-1	0	0
IGF 1	AZD 6244	t	1	0	1	-1	1	1
0,1% BSA	DMSO	c	-1	0	-1	-1	0	0
TGF_	AZD 6244	t	-1	1	-1	-1	1	1
0,1% BSA	DMSO	c	1	-1	-1	-1	-1	-1
0,1% BSA	LY294002	t	0	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	0	0	-1	-1	0
IGF 1	LY294002	t	1	1	1	-1	-1	1
0,1% BSA	DMSO	c	1	0	-1	-1	0	0
TGF_	LY294002	t	0	1	-1	-1	1	1

Fig. A.3. Perturbation data set from Klinger et al. [54] for cell line HCT116. A detailed description of the experiment, see Figure A.2. Rows marked in grey are data sets that caused a conflict and were removed.

SW480								
stimulator	inhibitor	type	Akt	Erk	GSK3	IRS-1	Mek	p70-S6K
0,1% BSA	DMSO	c	253	828	336	150	4948	1378
0,1% BSA	DMSO	c	264	869	278	126	4133	1309
0,1% BSA	DMSO	c	240	705	178	110	4070	1139
0,1% BSA	DMSO	c	513	1098	200	138	4550	1267
0,1% BSA	AZD 6244	t	225	410	163	129	3471	199
0,1% BSA	Ly 294002	t	108	1027	152	119	4972	1507
0,1% BSA	Ly 294002	t	95	1001	208	139	4900	1794
IGF 1	AZD 6244	t	23460	517	425	121	2392	1282
IGF 1	AZD 6244	t	24124	634	878	173	6620	2952
IGF 1	DMSO	t	24271	1330	1841	177	5331	3854
IGF 1	DMSO	t	24387	1818	518	168	5130	3424
IGF 1	DMSO	t	24538	1274	2039	184	5175	3078
IGF 1	DMSO	t	24247	1393	340	155	5291	2714
IGF 1	SB 216763	t	24838	1658	1086	219	5712	4439
IGF 1	SB 216763	t	24199	1589	1425	228	5721	3824
TGFa	AZD 6244	t	14168	2051	1883	223	15804	3121
TGFa	AZD 6244	t	15491	1247	383	182	13893	3132
TGFa	DMSO	t	10005	2942	531	319	13109	3779
TGFa	DMSO	t	10218	3597	415	188	11174	2782
TGFa	DMSO	t	6996	3268	548	360	15220	3326
TGFa	DMSO	t	9653	4257	548	232	13572	3563
TGFa	Ly 294002	t	133	2652	969	239	12289	2769
TGFa	Ly 294002	t	123	2499	283	195	12662	2444
0,1% BSA	DMSO	c	318	875	248	131	4425	1273
IGF 1	DMSO	t	24361	1454	1185	171	5232	3268
0,1% BSA	DMSO	c	318	875	248	131	4425	1273
TGF_	DMSO	t	9218	3516	511	275	13269	3363
0,1% BSA	DMSO	c	318	875	248	131	4425	1273
0,1% BSA	AZD 6244	t	167	719	158	124	4222	853
0,1% BSA	DMSO	c	318	875	248	131	4425	1273
IGF 1	AZD 6244	t	23792	576	652	147	4506	2117
0,1% BSA	DMSO	c	318	875	248	131	4425	1273
TGF_	AZD 6244	t	14830	1649	1133	203	14849	3127
0,1% BSA	DMSO	c	318	875	248	131	4425	1273
0,1% BSA	LY294002	t	102	1014	180	129	4936	1651
0,1% BSA	DMSO	c	318	875	248	131	4425	1273
TGF_	LY294002	t	128	2576	626	217	12476	2607
0,1% BSA	DMSO	c	0	-1	0	-1	-1	0
IGF 1	DMSO	t	1	-1	1	-1	-1	1
0,1% BSA	DMSO	c	0	0	0	0	0	0
TGF_	DMSO	t	1	1	1	1	1	1
0,1% BSA	DMSO	c	-1	-1	-1	-1	-1	-1
0,1% BSA	AZD 6244	t	-1	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	-1	0	-1	-1	-1
IGF 1	AZD 6244	t	1	-1	1	-1	-1	-1
0,1% BSA	DMSO	c	0	-1	0	-1	0	0
TGF_	AZD 6244	t	1	-1	1	-1	1	1
0,1% BSA	DMSO	c	1	-1	-1	-1	-1	-1
0,1% BSA	LY294002	t	0	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	1	0	0	-1	0	0
TGF_	LY294002	t	0	1	1	-1	1	1

Fig. A.4. Perturbation data set from Klinger et al. [54] for cell line HCT116. A detailed description of the experiment, see Figure A.2.

HT29								
stimulator	inhibitor	type	Akt	Erk	GSK3	IRS-1	Mek	p70-S6K
0,1% BSA	DMSO	c	213	879	113	208	2185	235
0,1% BSA	DMSO	c	252	990	140	244	2162	290
0,1% BSA	Medium	c	282	1083	122	254	2230	301
0,1% BSA	Medium	c	207	1393	171	324	2565	375
IGF 1	DMSO	t	4480	1419	239	232	3206	837
TGF_	DMSO	t	665	5245	378	826	5136	2358
0,1% BSA	AZD 6244	t	280	147	120	179	3287	227
0,1% BSA	AZD 6244	t	249	119	95	199	3846	203
IGF 1	AZD 6244	t	9814	176	545	275	4422	2300
TGF_	AZD 6244	t	932	4937	378	581	10891	2373
0,1% BSA	LY294002	t	119	940	94	204	2083	302
0,1% BSA	LY294002	t	112	898	99	202	1975	265
IGF 1	LY294002	t	200	1622	156	214	3148	519
TGF_	LY294002	t	155	7001	480	806	5906	2269
0,1% BSA	DMSO	c	239	1086	137	258	2286	300
IGF 1	DMSO	t	4480	1419	239	232	3206	837
0,1% BSA	DMSO	c	239	1086	137	258	2286	300
TGF_	DMSO	t	665	5245	378	826	5136	2358
0,1% BSA	DMSO	c	239	1086	137	258	2286	300
0,1% BSA	AZD 6244	t	265	133	108	189	3567	215
0,1% BSA	DMSO	c	239	1086	137	258	2286	300
IGF 1	AZD 6244	t	9814	176	545	275	4422	2300
0,1% BSA	DMSO	c	239	1086	137	258	2286	300
0,1% BSA	LY294002	t	116	919	97	203	2029	284
0,1% BSA	DMSO	c	239	1086	137	258	2286	300
IGF 1	LY294002	t	200	1622	156	214	3148	519
0,1% BSA	DMSO	c	239	1086	137	258	2286	300
TGF_	LY294002	t	155	7001	480	806	5906	2269
0,1% BSA	DMSO	c	0	-1	-1	-1	-1	0
IGF 1	DMSO	t	1	-1	-1	-1	-1	1
0,1% BSA	DMSO	c	0	0	0	0	0	0
TGF_	DMSO	t	1	1	1	1	1	1
0,1% BSA	DMSO	c	-1	1	-1	-1	-1	-1
0,1% BSA	AZD 6244	t	-1	0	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	1	0	-1	-1	0
IGF 1	AZD 6244	t	1	0	1	-1	-1	1
0,1% BSA	DMSO	c	1	-1	-1	-1	-1	-1
0,1% BSA	LY294002	t	0	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	-1	-1	-1	-1	-1	-1
IGF 1	LY294002	t	-1	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	-1	0	0	0	0	0
TGF_	LY294002	t	-1	1	1	1	1	1
0,1% BSA	DMSO	c	0	-1	-1	-1	-1	0
IGF 1	DMSO	t	1	-1	-1	-1	-1	1
0,1% BSA	DMSO	c	0	0	0	0	0	0
TGF_	DMSO	t	1	1	1	1	1	1
0,1% BSA	DMSO	c	-1	1	-1	-1	-1	-1
0,1% BSA	AZD 6244	t	-1	0	-1	-1	-1	-1
0,1% BSA	DMSO	c	0	1	0	-1	-1	0
IGF 1	AZD 6244	t	1	0	1	-1	-1	1
0,1% BSA	DMSO	c	-1	0	0	0	0	0
TGF_	AZD 6244	t	-1	1	1	1	1	1
0,1% BSA	DMSO	c	1	-1	-1	-1	-1	-1
0,1% BSA	LY294002	t	0	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	-1	-1	-1	-1	-1	-1
IGF 1	LY294002	t	-1	-1	-1	-1	-1	-1
0,1% BSA	DMSO	c	-1	0	0	0	0	0
TGF_	LY294002	t	-1	1	1	1	1	1

Fig. A.5. Perturbation data set from Klinger et al. [54] for cell line HCT116. A detailed description of the experiment, see Figure A.2. Rows marked in grey are data sets that caused a conflict and were removed.

	MZ1851RC-Sora						MZ1851RC-Sora						MZ1851RC-DMSO						MZ1851RC-DMSO					
	Erk 1/2	EGFR	mTOR	p70	S6	Akt	PTEN	VEGFR-2	p90 RSK	IGF-1R	Akt	Erk 1/2	EGFR	mTOR	p70	S6	Akt	PTEN	VEGFR-2	p90 RSK	IGF-1R	Akt		
20m	307.5	118	154	230	1296	540.5	137	29	40	176	53	737	223	326	498.5	4480	2468.5	241	27	62	221	63		
40m	460.5	169	239	344	3391	1941	218	26	39	210	68	664	223	312	449.5	3157.5	2572	213	27.5	54	227.5	64		
1h	518	200	278	386.5	2945	2504	234.5	27	44	247	75	601	161	232	328	3261	1850.5	191	25	53	178	63		
1.5h	452	160	221	327.5	1349.5	2289	211	26	45	195.5	62.5	508	202	229	329	NaN	1658	232	26	43	202	65		
2h	455.5	146	212	319	2661	1861	184.5	28	52	223	59	439	156	215	296	2394	1552	198	28	37	162	60.5		
4h	255	101	126	205	1777	820	195.5	28	37	181.5	58	350	127	157	236	3287	874	186	26	33	135	53		
8h	232	84	129	161	1759.5	517.5	197	29	34.5	181	50	221.5	102.5	113	185	1543.5	476	179	28	36	137	52		
12h	200.5	70	112	152	455.5	623.5	215	21	33	139	50	217	91.5	125.5	184	706	600	197	23	23	117	43		
24h	22	42	65	16	71	39	111	27	31	59	45	84	60	95.5	80	1053	879.5	167	22	29	90	49		
36h	23.5	40.5	24	15	60.5	38	35	27	24	52	44	61	53	97	72	NaN	286	282	23	26	125	46.5		
36h - untreated	65	48	108	70	64	200	386	24	26.5	165	47	65	48	108	70	64	200	386	24	26.5	165	47		
Mean	315	119	167	225	1789	1127	209	26	38	163	55	315	119	167	225	1789	1127	209	26	38	163	55		
Std.Deviation	217	61	81	139	1346	873	76	2	11	53	9	217	61	81	139	1346	873	76	2	11	53	9		
20m	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
40m	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1h	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1.5h	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
2h	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
4h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
8h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
12h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
24h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
36h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
36h - untreated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
20m	353.5	136	458	278	3009.5	569	222	22	69	48	42	491	178.5	542	355	3513	659	18	20	13	58	49		
40m	157	81	309.5	149	2779.5	444.5	205.5	20	64	49	40	383.5	143.5	400	305	3223	787	209	23	83	51	46		
1h	160.5	83	286	142	1929	365	196	22	57	44	42	386	138.5	459	308	3367	665	200	23	80	49	42		
1.5h	204	128	349.5	231	1481	430	169.5	22	47	50	44	387	155	497	350	4097	817.5	182	20	60	48	42.5		
2h	253	160	397.5	290	1711	557	156.5	21.5	48	46	39	379	152.5	388	357.5	3701	847	170	24.5	80	51	41		
4h	196	139	295.5	205	1691	411	160	23	38	44	41	446.5	188.5	436	424	1923	973	186	24	70	49	42		
8h	288.5	224	375	335	1127.5	434	186	27	43	55	46	346	153.5	343.5	313	2355.5	468	166.5	26	49	56	44		
12h	67	67	169	94.5	167	115	168	21	31	51	42	264	137.5	297.5	279	683	449.5	154.5	20.5	37	48	40		
24h	73	67	235	84	602	345.5	297	20	78	53	41	168	103	256	176	893	411	183.5	19	47	44	42		
36h	39.5	56	149	49	122	82	248	20	90	48	42	156	127	267.5	214.5	724	169.5	193	18	36.5	42	36		
36h - untreated	92	54	198.5	117.5	283	86.5	262	22	37	45	38	92	54	198.5	117.5	283	86.5	262	22	37	45	38		
Mean	233	121	322	230	1722	453	199	22	56	48	41	233	121	322	230	1722	453	199	22	56	48	41		
Std.Deviation	136	47	109	106	1280	259	54	2	20	4	3	136	47	109	106	1280	259	54	2	20	4	3		
20m	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1		
40m	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0		
1h	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0		
1.5h	0	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0		
2h	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0		
4h	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0		
8h	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0		
12h	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0		
24h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
36h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
36h - untreated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Fig. A.6. Data set from Christina Kuznia, showing the activity of MAPK and PI3K signaling over time using the Bio-Plex® platform. Cell lines MZ1257RC and MZ1851RC were treated with DMSO or Sorafenib and then measured from 10 minutes to 36 hours. The measurements were discretized by the mean value. The IGF-1R measurements marked red were excluded, due to the small variation in the data.

	Edges	Sora	Raf	Sora	EGFR	Sora	IGFR	EGFR	PI3K	Erk	mTr	mTr	IGFR	PI3K	Raf	Size	
a	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1 0.0%	
	2	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1 0.0%	
	2	0	0	0	0	1	0	0	0	1	0	0	1	0	0	2 0.1%	
b	2	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1 0.0%	
	2	0	0	0	1	0	0	0	1	0	0	1	0	0	0	2 0.1%	
	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1 0.0%	
	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1 0.0%	
	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1 0.0%	
	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	2 0.1%	
	2	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1 0.0%	
	2	0	1	0	0	1	0	0	0	0	0	0	0	0	0	2 0.1%	
	2	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1 0.0%	
	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1 0.0%	
	3	0	0	0	1	0	0	0	1	1	2	0	1	1	0	1 2.1%	
	3	0	0	0	1	0	0	0	1	1	4	0	1	1	0	1 4.1%	
	3	0	0	0	1	0	1	0	0	1	1	0	1	0	0	1 1.0%	
	3	0	0	0	1	0	1	0	1	0	1	0	1	0	0	1 1.0%	
	3	0	0	0	1	0	1	1	0	1	0	1	1	0	0	1 0.0%	
	3	0	0	0	1	0	1	1	0	1	1	0	1	0	0	2 0.1%	
	3	0	1	0	0	0	1	0	0	1	1	0	1	0	0	1 1.0%	
	3	0	1	0	0	0	1	0	0	1	3	0	1	0	0	1 3.1%	
	3	0	1	0	0	0	1	0	0	1	4	0	1	0	0	1 4.1%	
	3	0	1	0	0	0	1	1	0	0	1	0	1	0	0	1 0.0%	
	3	0	1	0	0	0	1	1	0	0	2	0	1	0	0	1 0.0%	
	3	1	0	0	0	0	1	0	0	1	4	0	1	0	0	1 4.1%	
	3	1	0	0	0	1	0	0	0	1	7	0	0	1	0	1 7.2%	
	3	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1 0.0%	
	4	0	0	0	1	0	1	1	1	1	1	0	1	1	0	1 1.0%	
	4	0	0	0	1	0	1	1	1	1	3	0	1	1	0	1 3.1%	
	4	0	0	0	1	0	1	1	1	1	6	0	2	1	0	1 6.2%	
4	0	0	0	1	0	1	1	1	1	1	0	1	1	0	1 1.0%		
4	0	0	1	0	0	1	1	1	1	1	0	1	1	0	1 1.0%		
4	0	0	1	0	0	1	1	1	1	2	0	1	1	0	1 2.1%		
4	0	0	1	0	0	1	0	1	1	4	0	1	1	0	1 4.1%		
4	0	0	1	0	1	0	1	0	1	8	0	2	1	0	1 8.2%		
4	0	1	0	0	1	1	0	1	1	5	0	1	0	0	1 5.1%		
4	0	1	0	0	1	1	0	1	1	12	0	3	1	0	1 12.3%		
4	0	1	0	0	1	1	1	0	1	6	0	2	1	0	1 6.2%		
4	0	1	0	0	1	1	0	1	0	6	0	2	1	0	1 6.2%		
4	0	1	0	0	1	1	1	0	0	5	0	1	0	0	1 5.1%		
4	0	1	0	0	1	1	1	1	0	10	0	3	1	0	1 10.3%		
4	1	0	0	0	1	1	0	1	1	12	0	3	1	0	1 12.3%		
5	0	0	0	1	1	1	1	1	1	3	0	1	1	0	1 3.1%		
5	0	0	0	1	1	1	1	1	1	3	0	1	1	0	1 3.1%		
5	0	0	0	1	1	1	1	1	1	6	0	2	1	0	1 6.2%		
5	0	0	0	1	1	1	1	1	1	3	0	1	1	0	1 3.1%		
5	0	1	0	0	1	1	1	1	1	9	0	2	1	0	1 9.2%		
5	0	1	0	0	1	1	1	1	1	18	0	5	1	0	1 18.5%		
c	3	0	1	0	1	0	0	1	0	0	0	1	0	0	0	8 0.2%	
	3	0	1	1	1	0	0	1	0	0	0	1	0	0	0	4 0.1%	
	3	0	1	1	1	0	1	0	0	0	0	1	0	0	0	1 0.0%	
	3	0	1	1	1	0	1	0	0	0	0	0	0	0	0	2 0.1%	
	3	0	1	1	1	0	1	0	0	0	0	0	0	0	0	3 0.1%	
	3	0	1	1	1	0	0	0	0	0	0	0	0	0	0	6 0.2%	
	3	0	1	1	1	1	0	0	0	0	0	0	0	0	0	4 0.1%	
	3	0	1	1	1	1	0	0	0	0	0	0	0	0	0	4 0.1%	
	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	4 0.1%	
	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	4 0.1%	
	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	4 0.1%	
	3	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	6 0.2%
	3	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	2 0.1%
	3	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	1 0.0%
	3	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	1 0.0%

Fig. A.7. Classification of the full model pool of Rp.1257 for all optional edges, where **a** shows model without Sorafenib target, **b** for one target, **c** for two target connections observable.

a													b																		
Edges	Sora	Raf	Sora	EGFR	Sora	IGFF	EGFR	PI3K	Erk	mTor	mTor	IGF	PI3K	Raf	Size	Edges	Sora	Raf	Sora	EGFR	Sora	IGFF	EGFR	PI3K	Erk	mTor	mTor	IGFF	PI3K	Raf	Size
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1 0.1%	3	1	0	0	0	0	0	1	0	0	0	0	1	2 0.2%		
1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1 0.1%	3	1	0	0	0	1	0	0	1	0	1	0	1	7 0.8%		
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1 0.1%	3	1	0	0	0	1	0	1	0	1	0	1	0	1 0.1%		
2	0	0	0	0	0	0	0	0	0	0	1	1	1	2 0.2%	3	1	0	0	0	1	0	1	0	1	0	1	0	1 0.1%			
2	0	0	0	0	0	0	0	0	0	0	1	1	1	2 0.2%	3	1	0	0	0	1	0	1	0	1	0	1	0	3 0.3%			
2	0	0	0	0	0	0	0	0	1	1	1	1	0	2 0.2%	3	1	0	0	0	1	0	1	0	1	0	1	0	3 0.3%			
2	0	0	0	0	0	0	1	0	1	1	1	1	0	2 0.2%	3	1	0	0	0	1	1	1	0	0	0	0	1 0.1%				
2	0	0	0	0	1	0	0	1	0	0	1	2 0.2%	3	1	0	0	1	1	0	0	1	0	0	0	0	1 0.1%					
2	0	0	0	0	1	0	1	0	1	0	0	1 0.1%	3	1	0	0	1	1	0	0	1	0	0	0	0	1 0.1%					
2	0	0	0	0	1	0	1	0	1	0	0	3 0.3%	4	0	0	1	0	1	1	1	1	1	1	1	1	1	1 0.1%				
2	0	0	0	0	1	0	1	0	1	0	0	1 0.1%	4	0	0	0	1	0	1	1	1	1	1	1	1	1	1 0.1%				
2	0	0	0	0	1	1	0	0	0	1	1	1	4 0.5%	4	0	0	0	1	1	1	0	1	1	1	1	1	1	1 0.1%			
2	0	0	0	0	1	1	0	0	1	1	1	1	4 0.5%	4	0	0	0	1	1	1	1	1	1	1	1	1	1	1 0.1%			
2	0	0	0	0	1	1	0	1	1	1	0	1	6 0.7%	4	0	0	0	1	1	1	1	1	1	1	1	1	0	13 1.5%			
2	0	0	0	0	1	1	0	1	1	1	0	1	10 1.1%	4	0	0	0	1	1	1	1	1	1	1	1	0	1 0.1%				
2	0	0	0	0	1	1	0	1	1	1	0	1	3 0.3%	4	0	1	0	0	1	0	1	1	1	1	1	1	1	2 0.2%			
2	0	0	0	0	1	1	1	1	1	1	1	0	2 0.2%	4	0	1	0	0	1	1	1	1	1	1	1	1	1	8 0.9%			
2	0	0	0	0	1	1	1	1	1	1	1	0	6 0.7%	4	0	1	0	1	0	1	0	1	1	1	1	1	1	1 0.1%			
2	0	0	0	0	1	1	1	1	1	1	1	1	4 0.5%	4	0	1	0	1	0	1	0	1	1	1	1	1	1	5 0.6%			
2	0	0	0	0	1	1	1	1	1	1	1	1	12 1.4%	4	0	1	0	1	0	1	1	1	1	1	1	0	2 0.2%				
2	0	1	1	0	0	0	0	0	0	1	0	0	2 0.2%	4	1	0	0	0	0	0	1	1	1	1	1	1	1	2 0.2%			
2	1	0	0	0	0	0	0	0	0	1	0	1 0.1%	4	1	0	0	0	0	0	1	1	1	1	1	1	1	11 1.2%				
2	1	0	0	0	0	0	0	0	0	1	0	1 0.1%	4	1	0	0	0	0	1	1	1	1	1	1	1	1	1	2 0.2%			
2	1	0	0	0	0	0	0	0	0	1	0	1 0.1%	4	1	0	0	0	0	1	0	1	1	1	1	1	1	1	14 1.6%			
2	1	0	0	0	1	0	0	0	0	1	0	0	1 0.1%	4	1	0	0	0	1	0	1	0	1	1	1	1	1	2 0.2%			
2	1	0	0	0	1	0	0	0	0	1	0	0	1 0.1%	4	1	0	0	0	1	0	1	0	1	1	1	1	1	7 0.8%			
2	1	0	0	0	1	0	0	0	0	1	0	0	1 0.1%	4	1	0	0	0	1	0	1	0	1	1	1	1	1	6 0.7%			
3	0	0	1	0	1	0	1	1	1	1	0	1	0	1 0.1%	4	1	0	0	0	1	0	1	0	1	1	1	1	21 2.4%			
3	0	0	1	0	1	0	1	1	1	1	0	0	3 0.3%	4	1	0	0	0	1	1	1	0	1	1	0	1	1	2 0.2%			
3	0	0	1	1	1	0	1	0	1	0	1	0	4 0.5%	4	1	0	0	0	1	1	1	0	1	1	0	1	1	2 0.2%			
3	0	0	1	1	1	1	0	1	0	1	0	0	1 0.1%	4	1	0	0	0	1	1	1	1	1	1	0	1	1	12 1.4%			
3	0	0	1	1	1	1	0	0	1	0	0	1 0.1%	4	1	0	0	0	1	1	1	1	1	1	1	0	1 0.1%					
3	0	1	0	0	0	0	0	1	1	1	1	1 0.1%	4	1	0	0	0	1	1	1	1	1	1	1	0	2 0.2%					
3	0	1	0	0	0	0	0	1	1	1	0	5 0.6%	4	1	0	0	0	1	1	1	1	1	1	1	0	3 0.3%					
3	0	1	0	0	0	0	1	1	1	0	2 0.2%	4	1	0	0	0	1	1	1	1	1	1	1	0	6 0.7%						
3	0	1	0	0	1	0	1	0	1	0	2 0.2%	5	0	0	1	1	1	1	1	1	1	1	1	1	1	1	2 0.2%				
3	1	0	0	0	0	0	0	0	1	1	2 0.2%	5	0	0	1	1	1	1	1	1	1	1	1	1	1	1	21 2.4%				
3	1	0	0	0	0	0	0	1	1	1	7 0.8%	5	0	0	1	1	1	1	1	1	1	1	1	1	1	1	3 0.3%				
3	1	0	0	0	0	0	0	1	1	2 0.2%	5	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	2 0.2%				
3	1	0	0	0	0	0	0	1	1	7 0.8%	5	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	8 0.9%				
3	1	0	0	0	0	1	1	0	1 0.1%	5	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	2 0.2%				
3	1	0	0	0	0	1	1	0	2 0.2%	5	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	11 1.2%				
3	1	0	0	0	0	1	1	0	1 0.1%	5	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	6 0.7%				
3	1	0	0	0	0	1	1	0	2 0.2%	5	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	42 4.8%			
3	1	0	0	0	0	1	1	0	1 0.1%	5	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6 0.7%			
3	1	0	0	0	0	1	1	0	2 0.2%	5	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	42 4.8%			

Fig. A.9. Classification of the full model pool of Rp.1851 for all optional edges, where **a** shows model without Sorafenib target, **b** for one target connection observable.

	Edges	Sora	Raf	Sora	EGFR	Sora	IGFF	EGFR	PI3K	Erk	mTor	mTor	IGF	PI3K	Raf	Size
c	3	1		1		0		0		0		1		0	1	0.1%
	3	1	1	1		0		0		0		1		0	3	0.3%
	3	1		1		0		0		0		1		0	2	0.2%
	3	1	1	1		0		1		0		0		0	3	0.3%
	4	1		0		1		0		1		1		0	1	0.1%
	4	1	1	0		1		0		1		1		0	3	0.3%
	4	1		0		1		1		0		1		0	1	0.1%
	4	1	1	0		1		1		0		1		0	2	0.2%
	4	1		0		1		1		0		1		0	4	0.5%
	4	1	1	0		1		1		0		1		0	4	0.5%
	4	1		0		1		1		0		1		0	1	0.1%
	4	1	1	0		1		1		0		1		0	1	0.1%
	4	1		1		0		0		0		1		1	1	0.1%
	4	1	1	1		0		0		0		1		1	6	0.7%
	4	1		1		0		0		0		1		1	6	0.7%
	4	1	1	1		0		0		0		1		1	16	1.8%
	4	1		1		0		0		1		1		0	1	0.1%
	4	1	1	1		0		0		1		1		0	3	0.3%
	4	1		1		0		0		1		1		0	2	0.2%
	4	1	1	1		0		1		0		0		1	6	0.7%
	4	1		1		0		1		0		1		0	1	0.1%
	4	1	1	1		0		1		0		1		0	5	0.6%
	4	1		1		0		1		0		1		0	2	0.2%
	4	1	1	1		0		1		0		1		0	4	0.5%
	4	1		1		0		1		1		0		0	3	0.3%
	5	1		0		1		0		1		1		1	2	0.2%
	5	1	1	0		1		0		1		1		1	6	0.7%
	5	1		0		1		1		0		1		1	2	0.2%
	5	1	1	0		1		1		0		1		1	4	0.5%
	5	1		0		1		1		0		1		1	1	0.1%
	5	1	1	0		1		1		0		1		1	8	0.9%
	5	1		0		1		1		0		1		1	28	3.2%
	5	1	1	0		1		1		0		1		1	2	0.2%
	5	1		0		1		1		0		1		1	7	0.8%
	5	1	1	0		1		1		1		0		1	2	0.2%
	5	1		0		1		1		1		1		0	1	0.1%
	5	1	1	0		1		1		1		1		0	2	0.2%
	5	1		0		1		1		1		1		0	2	0.2%
	5	1	1	0		1		1		1		1		0	4	0.5%
	5	1		0		1		1		1		1		0	13	1.5%
d	5	1		0		1		0		1		1		1	1	0.1%
	5	1	1	0		1		0		1		1		1	0	1.0%
	6	1		1		1		1		1		0		1	1	0.1%
	6	1	1	1		1		1		1		0		1	1	0.2%
	6	1		1		1		1		1		0		1	12	1.4%
	6	1	1	1		1		1		1		0		1	18	2.0%
	6	1		1		1		1		1		1		1	0	0.7%
	6	1	1	1		1		1		1		1		1	9	1.0%
	7	1		1		1		1		1		1		1	1	0.1%
	7	1	1	1		1		1		1		1		1	2	0.2%
	7	1		1		1		1		1		1		1	12	1.4%
	7	1	1	1		1		1		1		1		1	18	2.0%

Fig. A.10. Classification of the full model pool of Rp.1851 for all optional edges, where **c** shows models for two Sorafenib targets and **d** for all three target connections observable.

Abstract

This thesis is a contribution to the field of systems biology, where complex processes such as metabolism, gene regulation, or immune responses are formulated as mathematical representations to gain a comprehensive view. In order to create such a representation, called model, main characteristics of the system need to be idealized and simplified, where different modeling formalisms require different levels of simplification. This level can be seen as a trade-off between losing details and the amount of necessary information to validate this model.

Often models are built even though there is not enough information about the biological system available, which is circumvented by making assumptions. In this thesis, an alternative approach is presented, where the lack of information is included as uncertainty in the system. This uncertainty is used as constraints to create not one but every possible model that lies within these constraints giving rise to a pool of models.

In our group, software for building and analyzing these model pools in form of logical models was available, thus my work focuses on the biological application of this approach. The main task was to define how biology is translated into the mathematical formalism, to identify which kind of biological questions can be addressed and to interpret the mathematical results for gaining new biological insight.

These tasks were collected in a toolbox and applied to three different signaling systems that are interesting for cancer research. I investigated the effect of mutations on a signaling process, connected two pathways with uncertain crosstalk and investigated the controversial regulation of a protein complex involved in metabolism and cancer signaling.

Zusammenfassung

Diese Arbeit ist ein Beitrag zur Systembiologie, welche biologische Prozesse als mathematisches Konstrukt abstrahiert, um einen ganzheitlichen Blick von komplexen Vorgängen wie des Metabolismus, der Genregulation oder der Immunantwort zu erfassen. Diese sogenannten Modelle sind idealisierte und vereinfachte Darstellungen des ursprünglichen Systems mit dem Ziel, die Hauptcharakteristika zu erhalten. Dabei gibt es unterschiedliche mathematische Modellformalismen, welche es erlauben verschiedene Details der Biologie darzustellen, allerdings auch dementsprechend viele Informationen und Daten benötigen. Die Wahl des Formalismus ist also eine Abwägung zwischen Detailreichtum des Modells und der vorhandenen Datenlage.

Oft werden Modelle gebaut, obwohl nicht genügend Informationen vorhanden sind. In diesem Fall müssen Annahmen für die Unsicherheiten gemacht werden. Eine mögliche Alternative wird in dieser Doktorarbeit präsentiert, bei der diese Unsicherheiten als Bedingungen in das Modell integriert werden. Dadurch wird nicht nur ein Modell gebildet, sondern alle möglichen Modelle innerhalb dieser Bedingungen, so dass sich ein Modellpool ergibt.

In unserer Arbeitsgruppe wurde Software zur Generierung und Analyse für solche Pools von logischen Modellen entwickelt. In dieser Doktorarbeit wird dargestellt, wie biologische Information in diese Methodik eingebracht, verarbeitet und schließlich aus den Ergebnissen wieder extrahiert wird. Konkret wird untersucht, wie biologische Prozesse in den mathematischen Formalismus übersetzt werden, welche Fragestellungen mithilfe des Modellpools sinnvoll erörtert werden können und wie die mathematischen Ergebnisse dieser Methode als biologische Information interpretiert werden.

Diese drei Anwendungsbereiche werden in einer Toolbox zusammengetragen und auf verschiedene biologische Fragestellungen im Rahmen von Signalwegen in Krebszellen angewendet. Einerseits wurden mögliche Mutationen anhand von Hochdurchsatzdaten identifiziert, das Zusammenspiel von zwei einflussreichen Signalwegen in Nierenkrebszellen untersucht und die widersprüchliche Regulation eines Proteinkomplexes aufgeschlüsselt.

Ehrenwörtliche Erklärung

C

Hiermit erkläre ich, dass ich alle Hilfsmittel und Hilfen angegeben habe und versichere, auf dieser Grundlage die Arbeit selbständig verfasst zu haben. Die Arbeit wurde nicht schon einmal in einem früheren Promotionsverfahren eingereicht.

Berlin, März 2017

Kirsten Thobe

