Bipartite Boolean modelling - a method for mechanistic simulation and validation of large-scale signal transduction networks

Sebastian Thieme*, Jesper Romers*, Ulrike Münzner and Marcus Krantz

Institute for Biology, Humboldt-Universität zu Berlin, Berlin, Germany

* the authors contributed equally

Correspondence to: marcus.krantz@biologie.hu-berlin.de

Abstract

The metabolic modelling community has established the gold standard for bottom-up systems biology with reconstruction, validation and simulation of mechanistic genome-scale models. Similar methods have not yet been established for signal transduction networks, which appear to be much more challenging targets. This is due to the internal states carried by the signalling components, which encode and transmit the information through the network. Dealing with these states leads to scalability problems at two distinct level: First, in the model formulation, and second, in the model execution. For large-scale signalling networks, rule based modelling has been established as the quantitative method of choice due efficient model definition (as rules) and execution (as agents). However, rule based models cannot be simulated without quantitative parameters, introducing yet another layer of uncertainty. Consequently, it would be advantageous if models of signal transduction could be simulated and validated at the qualitative level. Here, we present a method for qualitative analysis of large-scale signalling networks. It is based on rxncon, the reaction-contingency language, a language for reconstruction of signal transduction networks. We developed a method to generate generic update rules for both the states and the reactions, the elementary building blocks in such a network, which can be used to map an arbitrary rxncon system on a unique bipartite Boolean model with a defined set of truth tables. Hence, each rxncon network can be converted into a corresponding executable model, providing a powerful tool for network validation. Furthermore, the rxncon network can also be compiled into a rule based model. Taken together, we equip rxncon 2.0 with a qualitative simulation tool, which can be used in network validation and model development.

Introduction

Systems biology aims towards the integrative analysis of large-scale biological systems up to whole cells. To realise this goal, we need to integrate knowledge into executable or computational models [1]. This process has been developed the furthest in the field of metabolic modelling, where the community routinely works with genome-scale models (GSMs). These models are defined at the level of biochemical reactions, cover the entire metabolic network of even complex cells, and can be simulated to predict system level functionality [2, 3]. The methodology is well established and has rich toolboxes for network reconstruction, validation and simulation [4], and constitutes the paradigm of bottom-up modelling. However, these tools cannot be used for signal transduction networks, due to the difference between mass and information transfer networks [5]. The main challenge in information transfer networks is to deal with the internal state changes of the components. Typically, signalling components undergo multiple state changes, leading to a combinatorial complexity problem that increases very fast with network scope [6, 7]. The combinatorial complexity and increase in the number of microstates within a signalling model, as well as the discrepancy in the available empirical data, increases with an increase in network size. This limits the size of signal transduction networks that can be simulated with reasonable computational costs. To overcome these issues, the community works on different strategies [8-10]. Of these strategies, rule based modelling holds the greatest promise for scalable quantitative mechanistic modelling.

To build a quantitative model, information is needed at several levels. The model information can be divided into a qualitative and a quantitative layer [2]. The qualitative layer requires information on components, states, reactions and contingencies, which define the model topology and regulation (in case of a signal transduction model; metabolic models typically ignore internal states (e.g. enzyme regulation by phosphorylation), despite rising evidence of their importance [11]). To turn the model structure into a quantitative model, rate laws and parameters are required. Consequently, modelling large networks requires data integration at many levels. For each level of detail (Components->States->Reactions->Contingencies->Rate Laws->Parameters), more and more data is required. At the same time, generating this data becomes increasingly difficult, leaving us with the lowest confidence at the highest (quantitative) model resolution. Hence, it would be valuable to enable simulation of these systems already at the qualitative level.

To enable simulation without parameters, the community developed several qualitative simulation methods. In particular, Boolean networks are commonly used. In these networks, model nodes represent e.g. biological molecules. The state of the nodes can either be TRUE or FALSE and this represents the activity of the molecules. The model states are updated in discrete steps, which provide a crude representation of time or at least causality. The changes in states over time are encoded in update rules that define how the state of one component at time t+1 depends on the state of the network at time t. These networks scale efficiently and can be executed for large networks. However, standard Boolean networks lack mechanistic detail and cannot distinguish between different active forms of a single component, which is highly problematic as many signalling components can be differentially activated against different targets (e.g. the cyclin dependent kinase). To address this, different groups developed mechanistic Boolean models [12-14], which can be used for a detailed description of signalling events through the combination of rule-based and site-specific logical modelling. These included a bipartite Boolean model formalism that described the network in terms of elemental reactions and state, which allowed it to retain the mechanistic

information by distinguishing between the different downstream functions of components. This enables us to approximate the underlying dynamics of the reconstructed network in a qualitative manner.

This bipartite Boolean modelling approach was based on rxncon, the reaction-contingency language. The rxncon language is tailored for the reconstruction of signal transduction networks [15]. As described in detail elsewhere [16], rxncon describes the network in terms of the mechanistic building blocks required for signalling: Elemental reaction that describe independent reaction in terms of changing elemental states, and their contingencies, the necessary molecular context of the reactions, expressed in terms of (combinations of) elemental states. The key feature is that a rxncon state is a macroscopic state. Except for the information on e.g. a phosphorylation, all other information is ignored; making the approach very similar to rule based modelling [17, 18]. Hence, the network definition corresponds very closely to empirical data and largely avoids the combinatorial complexity problem. This makes rxncon highly scalable as well as composable, as reaction and contingency statements are independent of each other. With rxncon 2.0, the second generation rxncon language, the expressiveness and precision is now on level with a rule based modelling language, and the rxncon 2.0 network definition can be compiled into a rule based model (RBM) in the BNGL language (Romers et al., unpublished). However, the RBM requires parametrisation in terms of rate constants and initial amounts, quantities for which reliable values are scarce. Hence, it would be valuable to enable qualitative simulation of these networks.

Here, we present a qualitative simulation method for rxncon 2.0. Based on the new syntax and semantics, we have created a new modelling formalism, only loosely related to the previous incarnation. This new formalism is based on two generic update rules for states and reactions, which can be used to map any rxncon network on a specific bipartite Boolean model with uniquely defined truth tables. We show that these models meaningfully predict system level function, and this, together with the unique mapping, provides a powerful validation tool: rxncon models can be directly compiled into models that can be executed without further optimisation at the system level. In the context of rule based model development, this would allow testing and optimisation of the qualitative model layer before entering the more expensive parametrisation and simulation cycles. In the context of genome-scale modelling, it will enable validation and simulation of large-scale, perhaps even genome-scale, signal transduction networks.

Methods

The rxncon language

The rxncon language systematically separates the network information into reactions (what happens) and contingencies (under what conditions does it happen). The biological information is described on the molecular level reflecting the resolution of the empirical data. For this, we define elemental states (a single property of a molecule, e.g. modification), elemental reactions (indivisible reaction events that synthesise, degrade, produce or consume elemental states) and contingencies (contextual constraints on elemental reaction). The term elemental means that the resolution of a state cannot be defined more precisely. Hence two states of the same type are defined on the same resolution e.g. residue level and if the residue is the same in both states they are mutually exclusive from each other. However, global quantities e.g. the pH value inside a cell does not fit the elemental concept, but can be described using Inputs and Outputs, which defined the boundary of the mechanistic model with the surrounding. Hence, global variables can be connected to the elemental states and reactions in the reaction-contingency formalism. The syntax and semantics of rxncon 2.0, the second generation rxncon language, is described in detail elsewhere [16]. For the generation of the bipartite Boolean model, the following features are of particular importance:

- The skeleton rule, which is a property of an elemental reaction, describing the transition between elemental states.
- The contingencies, holding the regulatory information.

The generic update rules

The information of the skeleton rule and the contingencies are sufficient to build the complete bipartite Boolean model. The skeleton rules provide the reaction layer of the system and therefore the information about which elemental state (represented by a state target) is produced, consumed, synthesised or degraded by an elemental reaction (represented by a reaction target). The contingencies provide the information how elemental reactions are regulated – the regulatory layer. From this we can construct the update rules for state targets and the update rules for reaction targets.

The generic reaction update rule consists of the components the reaction is based on and a Boolean combination of its contingencies. The regulatory layer does not include source states, hence, those are not included in the reaction target update rule.

The generic state target update rule integrates the effect of all elemental reactions on an elemental state. In a system without degradation and synthesis the truth value of a state target depends either on the presence of a production reaction for it and the source states for that production reaction. Alternatively, the state maintains the same value if it is not consumed by any reaction (which should in turn also have all its source states present). The full generic update rules, including degradation and synthesis reactions can be found in Fig S1.

Degradation reactions play a special role, because they are supposed to remove specific state targets or entire components from the system, depending on their regulation. Therefore, we have to consider two different pathways for integrating them into the system. If we have no regulatory

information on a degradation reaction target all state targets containing the respective component get degraded. If the degradation reaction target is regulated, the degradation itself will be targeted. Therefore, the regulatory information, which can be represented as a Boolean expression of regulatory states, has to be considered on top of the skeleton rules. This is done by building the disjunctive normal form (DNF), i.e. "OR of ANDs", of the contingencies. For each OR term within the solution a degradation reaction target is created and the respective regulatory information of the OR term is added to this reaction target (shown in Figure S2A). Special attention has to be given to mutually exclusive state targets. If the degradation reaction is inhibited by a state target, then this state target is protected for degradation, but its mutually exclusive state targets can be degraded. The opposite holds for requirements. If a state is required for the degradation reaction target the state target gets degraded, but the mutually exclusive state targets are protected. Furthermore, depending on the state target type, degradation reaction targets can also produce certain states e.g. if an interaction state target is degraded, the binding partner get released back to the system (Figure S2B). For this step, the degradation reaction target has to be divided in as many reaction targets as there are valid interaction state targets for that particular degradation reaction. Taken together, due to the interpretation what degradation translates to within a Boolean system, the degradation reaction targets play a different, less straightforward, than all other reaction targets.

The "component terms" are part of the state and the reaction target update rule: e.g. a reaction target can only be active if its components are active and the regulatory constraints are fulfilled. The term for a component is defined as a certain Boolean combination of state targets that live on that particular component. We construct it by separating the states for a component into mutually exclusive groups, taking the OR within a group and the AND between groups. Certain components in the system carry no elemental states: for these we introduce additional state targets.

The model creation workflow (as implemented)

The bipartite Boolean model is constructed from a rxncon system in seven steps. The first five steps are for collecting the information needed and the last two for constructing the update rules for state and reaction targets. Step one calculates all the components of the systems to get the information which component can be defined by elemental states and which not. Step two calculates the regulatory information of reaction targets using the DNF of their contingencies. In step three to five the calculation of the degradation reaction is handled. The components which are influenced by the degradation target are added, the contingency information are applied and interaction reactions are considered. In the last preparation step the information of synthesised components is added to the respective synthesis reaction. If this information is available the calculation of the update rules, according to the generic update rules for state and reaction targets, is straightforward. The system is initialised per default with all neutral state targets are TRUE and all other targets are FALSE.

The bipartite Boolean model is exported into a BoolNet compatible file (*.boolnet). To fulfil naming restrictions of BoolNet and to keep the names unambiguous the exported names of reaction and state targets are abbreviated with *R* or *S* respectively followed by a consecutive numbering. Additionally, to the BoolNet file the rxncon export provides two additional csv files: one mapping file (ending with *_symbols.csv) mapping an abbreviation to its respective target and one file (ending with *_initial_vals.csv) with initial values for all targets of the system. Per default all neutral state targets and generic component targets are initialized as ON and all other targets as OFF.

Model simulation

The simulation is done with the R CRAN package BoolNet [19]. This package provides methods for different kinds of simulation, analysis and visualisation of Boolean networks. The files needed for the simulation are the following: Bipartite Boolean network ends with '.boolnet', the mapping of the abbreviations ends with '_symbols.csv' and the file containing the initial conditions of the system ends with '_initial_vals.csv'. After reading those files in the functions of the BoolNet package can be used to simulate the bipartite Boolean model.

Within a Boolean model, we expect the output to be responsive to the input. The sign of the dependence does not matter, and we can start with the input either on or off. We simulate the model until it reaches an attractor. If it is a point attractor, we use it as starting point for the next simulation but turn the Input signal into a truth value, activating the output target and simulate again until we reach another attractor. We iteratively change inputs and simulate to an attractor state until we reach an attractor we have already seen.

Results

The expected behaviour of a small reaction circuit

First, we defined the behaviour we expect from a generic reaction motif. For this purpose, we designed a minimal circuit consisting of two elemental states – one neutral (unphosphorylated) and one modified (phosphorylated) – that are targeted by up to four reactions: A phosphorylation reaction, a dephosphorylation reaction, a synthesis reaction and a degradation reaction (Fig 1A). Each of these reactions has a different effect on the two state nodes (Table 1). Due to the crude notion of time in Boolean system, we decided to think of each truth table as a local equilibrium that only depended on the variables in the update rule. First, we considered the system in the absence of reactions. Here, we do not expect the states to change over time and consequently they remain as initiated (Fig 1B). Second, we added different combinations of phosphorylation and dephosphorylation reactions. If only the phosphorylation reaction is active, we expect the component to end up fully phosphorylated (P=TRUE, 0=FALSE). Similarly, with only the dephosphorylation reaction active, we would expect the component to be fully dephosphorylated (0=TRUE, P=FALSE). However, if both reactions are active, we would expect both forms of the component to be present at equilibrium (0=TRUE, P=TRUE) (Fig 2). In each of these three cases, this requires the component to be present (0=TRUE and/or P=TRUE), but it does not depend on which form is present (Fig 1C). Third, we added a degradation reaction to the system. In this case, we expect the component to be completely depleted from the system, and hence (0=FALSE, P=FALSE) regardless of phosphorylation, dephosphorylation, or initial states (Fig 1D). Fourth and finally, we added a synthesis to our system. With the same reasoning as for phosphorylation *vs.* dephosphorylation, we expect a state that is both synthesised and degraded to be present at equilibrium. Furthermore, we expect modified states of synthesised components to be true if their producing reactions are also active, even if these states are subject to degradation. Hence, if synthesis is active and phosphorylation is active, we expect both states to be TRUE even in the presence of degradation and/or dephosphorylation and regardless of initial conditions (0=TRUE, P=TRUE) (Fig 1E). In case of synthesis is active but not phosphorylation, there is no generation of the phosphorylated state and it is only stable in the absence of dephosphorylation and degradation. If any of these reactions are active, or if it is initialised as FALSE, it will be FALSE. Hence, we can fully define the expected truth values of states for any combination of active reactions, for both neutral and modified elemental states.

Reaction	Effect on neutral state (0)	Effect on modified state (P)
Phosphorylation (P+)	Consume	Produce
Dephosphorylation (P-)	Produce	Consume
Synthesis (Synth)	Synthesise	N/A
Degradation (Deg)	Degrade	Degrade

Table 1: Reactions have different effect on states.

Generic update rules define a molecular bipartite Boolean model

Next, we proceeded to define a semantic mapping from rxncon to a Boolean model that fulfils this behaviour. The objective of the Boolean model is to represent a rule based model without any free parameters that provides a systems-level (as opposed to molecular) description. We decided to keep the bipartite nature of the rxncon network with separate nodes for elemental reactions and elemental states. The update rules for these two set of nodes are also based on two distinct sets of edges: The truth value of elemental states depends on reactions, as discussed above, and the truth table of elemental reactions depends on contingencies, i.e. regulatory constraints on reactions. Together, reaction and state nodes, and transition and contingency edges suffice to describe the regulatory structure of the system. However, to turn this regulatory structure into a Boolean model, we need to make certain assumptions. In particular, we make three major assumptions: First, we assume that the truth value of a state derives from a local equilibrium motif, which allowed us to derive the truth tables above. Second, we assume that a state is TRUE if it abundant enough to be functionally important, and, inversely, that if it is TRUE it is functionally important. That, in combination with the equilibrium assumption discussed above, means that turning a state off requires inhibiting the reaction(s) creating it. Third, we assume that any intersection of elemental sets is TRUE as long as all the individual elemental states are TRUE: this allows us to translate contingencies that are defined on individual molecules into systems-level quantities. With these assumptions, we can build generic update rules for elemental reactions and states.

To create the Boolean models, we translate the rxncon model into a set of reaction targets and a set of state targets. The state targets correspond to elemental states, i.e. bound state at specific domains, modification state at specific residues, or localisation to a specific compartment. These states can be neutral (0), referring to the native state and compartment, or they can be bound/modified/localised with/by/to specific components/modifications/compartments. In addition, we create generic component state targets for all components in the network that have no elemental state variables, to enable synthesis, degradation and modelling of kinases that carry no further structure. The reaction targets correspond to elemental reactions, although sometimes with additional nodes for degradation reactions as discussed in detail in the methods section. The reaction targets also hold the information needed to create the update rules: First, the effect they have on different states (Synthesise, Degrade, Produce, Consume; compare Table 1). This information is extracted from the skeleton rule in the reaction type definition. Second, the regulatory constraints they are subject to, in the form of a Boolean expression of elemental states. The first formalises the reaction layer and the second formalises the contingency layer of the rxncon model. Together, they suffice to create the complete set of update rules.

The reaction target updates reflect the regulatory layer of the rxncon model. It directly corresponds to the contingency list. In a system without synthesis and degradation, the truth value of a reaction is TRUE if all required elemental states are TRUE and all inhibitory elemental states are FALSE. However, Boolean models can only account for all-or-nothing effect; hence quantitative contingencies are either mapped on qualitative contingencies or ignored. Hence, the reaction updates are straightforward to build. Note that the truth value of reactions is independent of the source state availability. A reaction being TRUE does not mean that it will execute, this also requires the source state to be present which is handled in the state update. This division is due to both technical and conceptual reasons. First, the reaction target update is intended to capture the contingency layer, which does not include source state information (this belongs to the reaction

layer). Second, it is necessary to avoid certain artefacts when we consider smoothing (see below). In a system with synthesis or degradation, we need to include the components in the update rule (see methods). In this general case, the reaction target is TRUE if all involved components are TRUE, all required elemental states are TRUE and all inhibitory elemental states are FALSE (Fig 3).

The state target updates reflect the reaction layer of the rxncon model. It integrates the effect of all elemental reactions on each elemental state into a truth value, in accordance with the reasoning above. First, it depends on the active reactions, in a hierarchy that depends on the effect each reaction has from the perspective of the specific target (compare table 1). Second, the effect of each reaction is conditional on the availability of their source state(s). That is, the reaction only has an effect if the reaction target is TRUE and all its source states, i.e. consumed states, are TRUE. In addition, a state target that is not subject to any reaction remains at its current truth value, i.e. it has a memory. To determine the truth value of a state target, the effect of the reactions is assembled with the following hierarchy: Synthesis > Degradation > Production > Consumption, in accordance with Figure 1. In a system without synthesis and degradation, and with one producing and one consuming reaction, the state will be TRUE if the producing reaction is TRUE and all its source states are TRUE, or if the state is TRUE and either the consuming reaction is FALSE or one of its source states is FALSE (Fig 3). Adding a synthesis and degradation reaction has different effects on the update rules for a neutral and a modified state, as only neutral states are synthesised directly. With degradation TRUE but synthesis FALSE, both a neutral and modified state is FALSE. However, a neutral state is always true with a synthesis reaction TRUE, while a modified state needs both a synthesis reaction to be TRUE and a producing reaction to be active (i.e. TRUE and with all source states TRUE), as the state is not directly synthesised. In addition, all components in the state must be TRUE, as in the reaction updates. Hence, state updates are more complex than reaction updates, but knowledge of the role each reaction has on each state target suffices for assembly of complete update rules.

Together, the reaction and state target updates completely define the bipartite Boolean model. To test if the update logic described above captures the expected behaviour, we implemented the model generation process (see methods) and used it to generate the 64 models corresponding to the minimal circuitry above (Fig 1A). The models were simulated using BoolNet [19], as described in detail in the methods. The attractor states are visualised in Figure 4. The model behaviours correspond to our expectations with one notable exception. In the absence of synthesis and degradation, but in the presence of both phosphorylation and dephosphorylation, the model displays an oscillatory behaviour when only one of the two states is initiated (Fig 4A). Closer inspection reveals that this is due to periodic source state depletion. The phosphorylation and dephosphorylation reactions are constitutive (no contingencies, no loss of components), and the oscillation due to the states are completely encoded in the state update rule. Indeed, as soon as a reaction executes, dependent on its source state, it depletes the source state pool. Hence, the reactions alternate in firing, triggering out-of-phase oscillation in the truth value of the states. Consistently, these oscillations disappear when both state are initiated or when the source state is repleted through synthesis. We consider these trivial oscillations highly undesirable in our systems-level description, but note that they would be appropriate for models of single molecules. This may be consistent with the rule based basis of the algorithm, but the trivial oscillations should be eliminated for the method to be useful for system level simulations. Nevertheless, the outcome is highly encouraging, as 62 out of 64 models matched the expected behaviour.

Source state smoothing eliminates the trivial oscillations

Next, we adapted the bBM logic to capture the signal transduction network at the single cell level. To do this, we eliminated the trivial source state oscillations in reaction cycles. The basic assumption behind this goes back to the assumption of local equilibrium: In a reaction cycle with both forward and reverse reactions, both states will be present at equilibrium. To implement this, we adapted the state target updates by widening the window in which we checked for source state availability: A reaction now needs the source state to be present, or to be produced (Fig 3C). This leads to a situation where states that are mutually exclusive at the single molecule level can be present at the same time at the systems level. This can be justified by integrating over molecules or over time. If many molecules are present in the system, and if both reactions in the cycle are active, then both of them will be present in some molecules at any given time. Similarly, if a single molecule is cycling through two mutually exclusive states sufficiently fast, then, within the crude unit of a Boolean time-step, the molecule will be present in both states within the time window. The smoothing assumption only breaks down in the context of few molecules and low reaction rates, but in this case the reaction could also be considered to be functionally off. Having established the smoothing logic, we implemented it into the rxncon compiler tool (see methods) and recreated the 64 models with smoothing. We repeated the simulation and compared the results to the original simulation (Fig 4B). The oscillatory behaviour disappeared, but no other simulation results changed. Hence, the simulation results exactly match the behaviour we expect from a model of this reaction motif.

The update rules can be used as LEGO bricks to assemble a systems level model

Next, we applied the bBM logic to simulate a linear pathway. We chose a simplified model of the HOG MAP kinase pathway from *Saccharomyces cerevisiae* (Fig 5; taken from [14]). We created a rxncon 2.0 model of this pathway (Table S1), and used this to generate the bBM using the generic update rules with smoothing. Already this small model has 28 reaction and state targets, and hence 2^{28} (~10^8) possible initial states. As this is too many for an exhaustive search, we decided to use a generic start state for all simulations: All neutral elemental states are TRUE, all generic component states (for components with no elemental states) are TRUE, and all other nodes are FALSE. From this highly artificial initial state, we let the model find its own natural “off-state” by executing it until an attractor is reached (Fig 5B). At this point, we change the input state and repeat to see the response of the pathway to the input, and repeat this process until the model returns to a state we have already seen. As can be seen from Figure 5B, the HOG pathway responds appropriately to turgor: It turns off the kinase cascade. For comparison, we repeated the simulation with the non-smoothed logic (Fig 5C), where we see the signal passing the network despite trivial oscillations. However, the system does not converge to point attractors, leading to more complex analysis and interpretation. There are three striking blocks in the heat-map: First, the initial neutral states never turn off. Second, there is a block of reactions that turn on directly, and stay on throughout the simulation. Third, there is a block that turns on and off in response to the signal. The third block contains the reactions and states that actually transmit the information. The second block contains constitutive reactions, which are either unregulated (e.g. dephosphorylation reactions), or at the level of source state availability (e.g. phosphotransfer from Sln1 to Ypd1). The first block contains all the neutral states. These remain true because the reactions that produce them are considered unregulated, which may

be due to experimental bias as discussed below. Hence, the logic of the generic update rules is sufficient to convert a rxncon network into a functional bBM that accurately predicts system level function. It is highly non-trivial that generic update rules that were defined for isolated reactions suffice to define a complete model that functions at the systems level, with no further tweaking or parametrisation. Taken together, the generic update rules map any given rxncon network on a unique Boolean model that predicts systems level function.

The bipartite Boolean logic correctly reproduces real oscillations

The HOG pathway is a homeostatic pathway that maintains proper turgor pressure. The pathway output eventually leads to signal cessation through a physiological feedback loop [20]. To simulate this, we linked the most downstream component to the input that turns the pathway off (Fig 5A). We repeated the model creation and simulation, using the initial steady state of the linear model as starting condition. As shown in Figure 5D, the model now shows a periodic activation/deactivation behaviour similar to that when the input is changed manually. Hence, the bBM logic is fully capable of predicting biologically relevant oscillations. As comparison, we also simulated the cyclic HOG model without source state smoothing (Fig 5E). Here, the pathway signal is almost completely washed out by the trivial oscillations. The source state smoothing facilitates bBM analysis and clearly improves the interpretability of the simulation results. Taken together, the bBM logic generates Boolean models that can predict systems level function for both linear and cyclic systems.

The bipartite Boolean logic scales to large-scale systems

Finally, we applied the method on the pheromone response pathway of baker's yeast. We chose this pathway to benchmark the bBM method due to the existence of an excellently annotated, comprehensive and detailed RBM [21]. We previously translated this model to rxncon [15], making it readily available for analysis. The original RBM contains 229 rules with 200 parameters (166 unknown) that define how 20 components can assume over 200.000 distinct states (http://yeastpheromonemodel.org/wiki/Extracting_the_model). While this is one of the most carefully built and curated RBMs, it remains difficult to meaningfully simulate it as such. Hence, it constitutes an excellent benchmark target for the bBM method.

We simulated the pheromone bBM using a standardised simulation workflow (see methods). The rxncon implementation of the yeast pheromone model contains 163 elemental reactions, corresponding to 103 elemental states and 190 contingencies (Table S2). In the translation, we added undefined catalysts as unknown components which increased the component number but otherwise did not affect the complexity of the model. We generated the bBM using the smoothed update rules, which produced a bipartite Boolean model 209 reaction targets (due to duplication of degradation reaction targets; see material and methods), 180 state targets (including component targets having no elemental states e.g. Ste7mRNA). With 389 state variables, the model is too large to use an exhaustive search of initial states (statespace = 2^{389}; ca 10^{117} distinct configurations), so we rely on the default initiation state (all neutral state targets are TRUE, all generic component targets TRUE, all other targets are FALSE). From this initiation state, we first let the model find its natural "off-state", as explained for the HOG pathway above, before we iteratively switched the input to TRUE and FALSE (Fig 6). We found that the pathway did not behave as expected, as the statements

in the RBM have different quantitative effects and this information is lost in the Boolean model. We thus turned a minimal set of quantitative contingencies (K+/K-) that were ignored in the model generation into qualitative contingencies (!/x). In total, we needed to change 3 of the 91 quantitative contingencies to explain information transfer, indicating that a simpler model would be able to explain the pathway functionality. The updated version of the model does respond to pheromone treatment as expected.

Taken together, these results show that the bipartite Boolean logic is scalable, and can be used to predict behaviour of systems that are too large to be simulated by rule based modelling. Furthermore, it highlights the value of the bBM approach in network validation: The generic update rules map rxncon models on a unique bBM that can predict the system level functionality of that rxncon model.

Discussion

Here, we present a qualitative simulation method for large-scale mechanistically detailed signal transduction networks. The formalism is based on Boolean logic and makes use of the BoolNet package for simulation. However, we present a completely new simulation logic. Based on detailed analysis of a simple reaction motif, and on a small set of standard assumptions, we define two generic update rules: one for reaction and one for state targets. These generic update rules map a bipartite rxncon network on a unique bipartite Boolean model with defined truth tables. The elemental reactions define the update rules for state targets, and the contingencies define the update rules for reaction targets. We show that these building blocks can be assembled like LEGO-bricks into a bipartite Boolean model that predicts system level function without further optimisation at the system level.

The unique mapping from rxncon to an executable bBM that predicts system behaviour is highly non-trivial, and provides a powerful tool for network validation. Normally, it is relatively easy to build a Boolean model structure, but highly non-trivial to define truth tables that enable the model to reproduce the behaviour of the system. Here, we find that the regulatory structure encoded in the rxncon network already uniquely defines a Boolean model with a set truth table, and that this Boolean model meaningfully predicts system level behaviour. This has far-reaching implications: First, it provides a computationally cheap validation tool in the process of building rule based models. The second generation rxncon is an expressive and precise language in which to build rule based models [16]. However, in contrast to direct rule based model development, the rxncon network can also be used to generate a bBM. As the bBM can be simulated directly without parametrisation, it provides a tool to validate the model structure without prior parametrisation. Hence, it can be used to finalise developing the structural part of the rule based model before entering the more expensive parametrisation cycles. Second, the rxncon language can be used to build large-scale, perhaps even genome-scale, models of signal transduction. The unambiguous mapping on an executable model provides a validation tool for these networks, similar to what exists for metabolic networks [4, 5]. Therefore, rxncon 2.0 can be used to create large-scale models of signal transduction, which can be directly validated with the bBM method presented here. The bBM formalises the molecular interactions at the microscopic system level into a model that can predict the pathway or network function at the macroscopic level (e.g. input-output relationships). Hence, the work presented here will contribute to a reconstruction workflow for large-scale signal transduction network.

The bipartite Boolean modelling formalism here is a complete reinvention of the bBM published previously [14]. First, it is based on the second generation rxncon language, which has eliminated the shortcomings of the initial rxncon language and created a syntactically and semantically well-defined language [16]. In particular, rxncon 2.0 encodes the semantic meaning of reactions in the reaction type, which is of particular importance to mapping the rxncon reactions on the generic update rules. Moreover, it includes neutral states and enables states that are mutually exclusive at the single molecule to be present simultaneously at the systems level. That many of these neutral states appear unregulated in the models we analyse may have two reasons: First, there could be a constant pool of unmodified components in the cell, which is especially likely if there is a constant turnover (and hence synthesis, which per definition occurs in the neutral states). Second, it could reflect an experimental bias: At least in yeast, we know much more about the modifying reactions than about the reactions that reverse the modification (e.g. phosphorylation vs dephosphorylation,

[15]). In addition to the change in the rxncon language itself, we have engineered the update rules from scratch. This has led to a number of important differences: First, the state updates mirror the reaction layer, including the dependence on source states for reactions. Second, synthesis and degradation of components are now mapped on elemental states, and degradation now functions both at the level of partial degradation (e.g. a ubiquitylated form of a protein is degraded, but the non-ubiquitylated pools remain protected) and at the level of complete degradation (all states and reactions involving a protein turn off). Third, the component states are not independent model entities, but a test for the presence for each residue or domain: If all mutually exclusive states at a single site or domain are FALSE, then the component has been completely degraded. Comparing the difference between the two formalisms, it is surprising that the original simulation method worked in so many cases [14, 22, 23]. Except a few issues around degradation (components degraded, but states/reactions remained true), the formalism worked well despite the bias towards modified states (neutral states were implicit) and the fact that modified and neutral states could not coexist even at the systems level. This may again be due to experimental bias: We simply know much more about the function of modifications. However, it may also be that the original rxncon language and bBM logic already was a useful approximation, describing the key features of the system. In either case, rxncon 2.0 and the new bBM logic provide a syntactically and semantically sounder foundation for large-scale network reconstruction, validation and simulation.

These tools enable a scale-shift in signal transduction modelling. Hitherto, signal transduction models have been large-scale or mechanistically detailed, but not both. There are good reasons for this, as it is difficult to scale these models up. In particular, the microstate description – as exemplified by SBGN-PD diagrams - has catastrophic scaling due the combinatorial complexity of complexes and modifications [6]. Indeed, the large-scale efforts made in this format, by e.g. the Reactome team, who maps human signal transduction in the SBGN-PD format, have to be broken down into smaller models that are individually of much smaller scale [24]. These problems are largely alleviated in rule based modelling languages, which have been defined to address the combinatorial complexity [17, 18]. The rule based approach supports efficient model definition (since the rules only contain the relevant centre and context and ignore the rest of the internal state of the reactants) and efficient model execution (as agent based models; [25]). However, updates and extensions can have very non-local impact on the set of rules. Hence, it will be more difficult to maintain and extend a rule-based model as the scope increases. In contrast, the rxncon language makes use of independent statements, where additions and revision only have local effect in that particular reaction, the state(s) it change, and the contingencies that directly use them [16]. Hence, the rxncon language enables large-, possibly genome-, scale signal transduction networks. These networks can be used to create the corresponding rule based networks, but those models still require parametrisation – which will remain infeasible at the genome-scale for the foreseeable future. Here, we present a method to simulate such networks without the need for parametrisation. The method is qualitative only, but even metabolic modelling – clearly the state-of-the-art in genome-scale modelling – is limited to qualitative or semi-quantitative methods at the genome scale. Taken together, the method we present here has the potential to enable simulation of genome-scale models of signal transduction. Together with the rxncon 2.0 language, it offers the possibility to build, validate and simulate genome-scale models of signal transduction networks – that can be turned into rule based models as soon as the quantitative knowledge makes it meaningful.

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research via e:Bio Cellemental (FKZ0316193, to MK).

Figure Legends

Figure 1: Update expectations from a generic reaction motif. Here, we show a minimal circuit with maximal four reactions and two elemental states of a generic protein. We show the expected equilibrium behaviour after systematically turning reactions and the initial states of the two elemental states of the protein on or off. (A) The minimal circuit: A generic protein is synthesised, degraded, phosphorylated and dephosphorylated. (B) In a system where no reactions take place, the equilibrium states will remain the same as the initial states. (C) In a system without protein turnover but phosphorylation, the protein will be fully phosphorylated. The phosphorylated state is absent in a system with only dephosphorylation. Analogously, with dephosphorylation active, the protein will be fully dephosphorylated. With both reactions active, we expect both elemental states to be present at equilibrium (compare Fig 2). (D) In a system where degradation but not synthesis is active, neither of the two elemental states will be present (the protein is completely depleted) regardless of the (de)phosphorylation reactions and initial states. (E) In a system with protein synthesis and phosphorylation, both elemental states will be true, regardless of the presence of the other reactions and the initial values. Without phosphorylation, only the unmodified protein will be in the system. The exception is the case where degradation and phosphorylation are inactive but the phosphorylated protein is present initially. As no reactions affect the phosphorylated protein, it will remain in the system (compare panel B).

Figure 2: Discretisation of rate equilibrium. The figure shows the comparison between the equilibrium of rates in a dynamic system and a qualitative approach. The inner graph shows the behaviour of a dynamic system with two different states; unphosphorylated (0) and phosphorylated (P), under two different ratios of phosphorylation to dephosphorylation (assuming first order kinetics): I) The rate constants are identical II) Phosphorylation is an order of magnitude faster. In both cases, we can observe that the concentration of one state (P) increases and the concentration of the other (0) decreases, but at equilibrium both states will be present. We note that the forward reaction is qualitatively dominant over the reverse reaction (from the perspective of each state). If a state is being produced by a reaction, the state will be true. We discretise the reaction system accordingly, assuming that both states will be true in a reaction cycle where both reactions are active.

Figure 3: The generic update rules. The figure shows the generic update rules of the bipartite Boolean Model for reaction targets and state targets. For simplicity, we add only one component and one reaction of each type. (A) shows a system without degradation and synthesis. The update rule for the reaction depends only on its contingencies. The Boolean expression of the contingency depends on the reconstruction. The state target update rule depends only on producing and consuming reactions and those on their source state targets. (B) Synthesis and degradation reactions have an influence on a component level on the system. To take this into account we add components to both update rules. A component is defined as a Boolean combination of state targets sharing a specific component. State targets which cannot coexist on a single molecule level are combined by Boolean OR and state targets which can coexist are combined by Boolean AND operators (see legend). Now, the update rule for a reaction target depends on a component and its contingencies. In the update rule for a state target we add a single component, synthesis and degradation reaction. Each of the state target update rules can be modified by smoothing. This is realised by replacing the source state target by the source state target OR the production of the

respective source state target AND the source states needed to run the reaction AND its synthesis OR NOT its degradation. This ensures that the system will have the capacity of running the reaction now or a step ahead.

Figure 4: The possible attractors of the generic reaction motif. The minimal reaction circuitry was implemented and simulated with each of the 16 possible reaction configurations from all four possible initial states. We compared the attractor state to the expectations (Fig 1). (A) Using the basic update rules, we cannot reproduce the expected behaviour in two of the 64 simulations. When both phosphorylation and dephosphorylation reactions are active, and one but not both states are initiated as TRUE, then the two states do not stabilise as TRUE but oscillate out of phase between TRUE and FALSE (arrows). (B) Using the source state smoothed update rules, we can reproduce the expected behaviour in all 64 simulations. The trivial oscillations are gone, and both states now stabilise as TRUE under these conditions.

Figure 5: A bipartite Boolean model of the HOG pathway. (A) The regulatory graph shows the Sln1 branch of the HOG pathway. In the absence of turgor, Sln1 stays unphosphorylated. As turgor increases, the autophosphorylation of Sln1 activates a phosphotransfer cascade converging at Ssk1. The phosphorylated form of Ssk1 turns off the downstream MAP kinase pathway leading to dephosphorylation of the downstream transcription factor Hot1. (B) Simulation of a linear version of the model using source state smoothing of the update rules. We use our default assumptions (I) on the initial state and simulate the model until we reach an attractor (first OFF trajectory). We activate the system by turning [Turgor] ON and simulate again (ON trajectory) until we reach an attractor state. From there, we set [Turgor] OFF again and simulate the model until we reach an attractor. We observe that the model responds as expected to the input. (C) Simulation of the linear model without smoothing. The signal goes through the pathway, but analysis is complicated by the trivial oscillations as the system no longer converges on point attractors.(D) We extended the HOG model with a feedback loop, where activation of the pathway leads to increased turgor (via Hot1-{P}). This is simplification of an adaptive response through increased glycerol production and retention, which increases turgor. We simulate this model form the initial OFF attractor (see panel B), and note that the system oscillates as expected. (E) Simulation of the cyclic HOG model without smoothing. While a higher-order oscillation is detectable, the original signal is all but completely lost under the trivial oscillations.

Figure 6: Simulation of the yeast pheromone pathway. The simulation results of the pheromone model displayed without synthesis and degradation reaction, for clarity. We show the model behaviour upon stimulation with pheromone. Adding pheromone leads to the activation of a MAPK cascade with the phosphorylation of Far1, Dig1 and Ste11 as the most downstream targets that we consider in this model. Upon removal of pheromone, the system eventually goes back to the original state, except for the feedback phosphorylation which – in the original model – is only eliminated by degradation.

Figure S1: The processing of degradation reactions. Degradation reactions differ in their interpretation depending on if they are regulated or not. (A) shows two different ways to integrate degradation reactions into the system. The left way considers the case that the reaction is not regulated. In this case, all state targets containing the respective component are considered for degradation. If the degradation reaction target is regulated, the regulatory information has to be

considered on top of the skeleton rules. In a first step, we are building the disjunctive normal form (DNF) of the contingencies. For each OR term within the solution a degradation reaction target is created and the respective regulatory information of the OR term is added to this reaction target. In a last step, interaction states have to be considered for the regulated as well as the not regulated degradation reaction targets. Since we have two different components in this state target type, degradation reactions will consume the state target but produce the unbound form of the binding partner of the degraded component. For this step, the degradation reaction target has to be divided in as many reaction targets as there are valid interaction state targets for that particular degradation reaction. (B) shows a simulation of the degradation depending on an interaction state. As soon as the interaction state is build the degradation reaction turns on and degrades the interaction state. During this process the not degraded component is released in its neutral unbound form.

Figure S2: The generic update rules for multiple components and reactions. A reaction target can depend on multiple components. A component is defined as a Boolean combination of state targets sharing a specific component. State targets which cannot coexist on a single molecule level are combined by Boolean OR and state targets which can coexist are combined by Boolean AND (see legend). If a reaction depends on multiple components they are combined by Boolean AND, whereas the combination of contingency state targets depends on the reconstruction of the regulatory mechanism. Similarly, a state target can be influenced by more than one reaction of the same type, e.g. produced or consumed by different reactions. Therefore, reactions of the same type can be combined by Boolean OR if they have a positive impact on the truth value e.g. synthesis and production reactions, whereas reaction targets with a negative impact e.g. degradation and consumption reactions are combined by Boolean AND of their negation. Note that a consumption reaction depends on its source state, which can also be multiple ones. The negation of the Boolean AND combination will result in a Boolean OR combination of the negated elements.

References

1. Fisher, J. and T.A. Henzinger, *Executable cell biology.* Nat Biotechnol, 2007. **25**(11): p. 1239-49.
2. Herrgard, M.J., et al., *A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology.* Nat Biotechnol, 2008. **26**(10): p. 1155-60.
3. Thiele, I., et al., *A community-driven global reconstruction of human metabolism.* Nat Biotechnol, 2013. **31**(5): p. 419-25.
4. Thiele, I. and B.O. Palsson, *A protocol for generating a high-quality genome-scale metabolic reconstruction.* Nat Protoc, 2010. **5**(1): p. 93-121.
5. Münzner, U., et al., *Towards genome-scale models of signal transduction networks.* accepted.
6. Hlavacek, W.S., et al., *The complexity of complexes in signal transduction.* Biotechnol Bioeng, 2003. **84**(7): p. 783-94.
7. Faeder, J.R., et al., *Combinatorial complexity and dynamical restriction of network flows in signal transduction.* Syst Biol (Stevenage), 2005. **2**(1): p. 5-15.
8. Faeder, J.R., M.L. Blinov, and W.S. Hlavacek, *Rule-based modeling of biochemical systems with BioNetGen.* Methods Mol Biol, 2009. **500**: p. 113-67.
9. Borisov, N.M., et al., *Domain-oriented reduction of rule-based network models.* IET Syst Biol, 2008. **2**(5): p. 342-51.
10. Samaga, R. and S. Klamt, *Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks.* Cell Commun Signal, 2013. **11**(1): p. 43.
11. Oliveira, A.P., et al., *Regulation of yeast central metabolism by enzyme phosphorylation.* Mol Syst Biol, 2012. **8**: p. 623.
12. Handorf, T. and E. Klipp, *Modeling mechanistic biological networks: an advanced Boolean approach.* Bioinformatics, 2012. **28**(4): p. 557-63.
13. Kolczyk, K., et al., *The Process-Interaction-Model: a common representation of rule-based and logical models allows studying signal transduction on different levels of detail.* BMC Bioinformatics, 2012. **13**: p. 251.
14. Flottmann, M., et al., *Reaction-contingency based bipartite Boolean modelling.* BMC Syst Biol, 2013. **7**: p. 58.
15. Tiger, C.F., et al., *A framework for mapping, visualisation and automatic model creation of signal-transduction networks.* Mol Syst Biol, 2012. **8**: p. 578.
16. Romers, J. and M. Krantz, *rxncon 2.0: a language for executable molecular systems biology.* in preparation.
17. Blinov, M.L., et al., *BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains.* Bioinformatics, 2004. **20**(17): p. 3289-91.
18. Danos, V., et al., *Rule-Based Modelling of Cellular Signalling*, in *CONCUR 2007 – Concurrency Theory: 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007. Proceedings*, L. Caires and V.T. Vasconcelos, Editors. 2007, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 17-41.
19. Mussel, C., M. Hopfensitz, and H.A. Kestler, *BoolNet--an R package for generation, reconstruction and analysis of Boolean networks.* Bioinformatics, 2010. **26**(10): p. 1378-80.
20. Klipp, E., et al., *Integrative model of the response of yeast to osmotic shock.* Nat Biotechnol, 2005. **23**(8): p. 975-82.

21. Thomson, T. *Yeast Pheromone Model*. Available from: http://yeastpheromonemodel.org.
22. Mori, T., et al., *Stochastic simulation of Boolean rxncon models: towards quantitative analysis of large signaling networks.* BMC Syst Biol, 2015. **9**: p. 45.
23. Lubitz, T., et al., *Network reconstruction and validation of the Snf1/AMPK pathway in baker's yeast based on a comprehensive literature review.* Npj Systems Biology And Applications, 2015. **1**: p. 15007.
24. Robertson, M., *Reactome: clear view of a starry sky.* Drug Discov Today, 2004. **9**(16): p. 684-5.
25. Sneddon, M.W., J.R. Faeder, and T. Emonet, *Efficient modeling, simulation and coarse-graining of biological complexity with NFsim.* Nat Methods, 2011. **8**(2): p. 177-83.

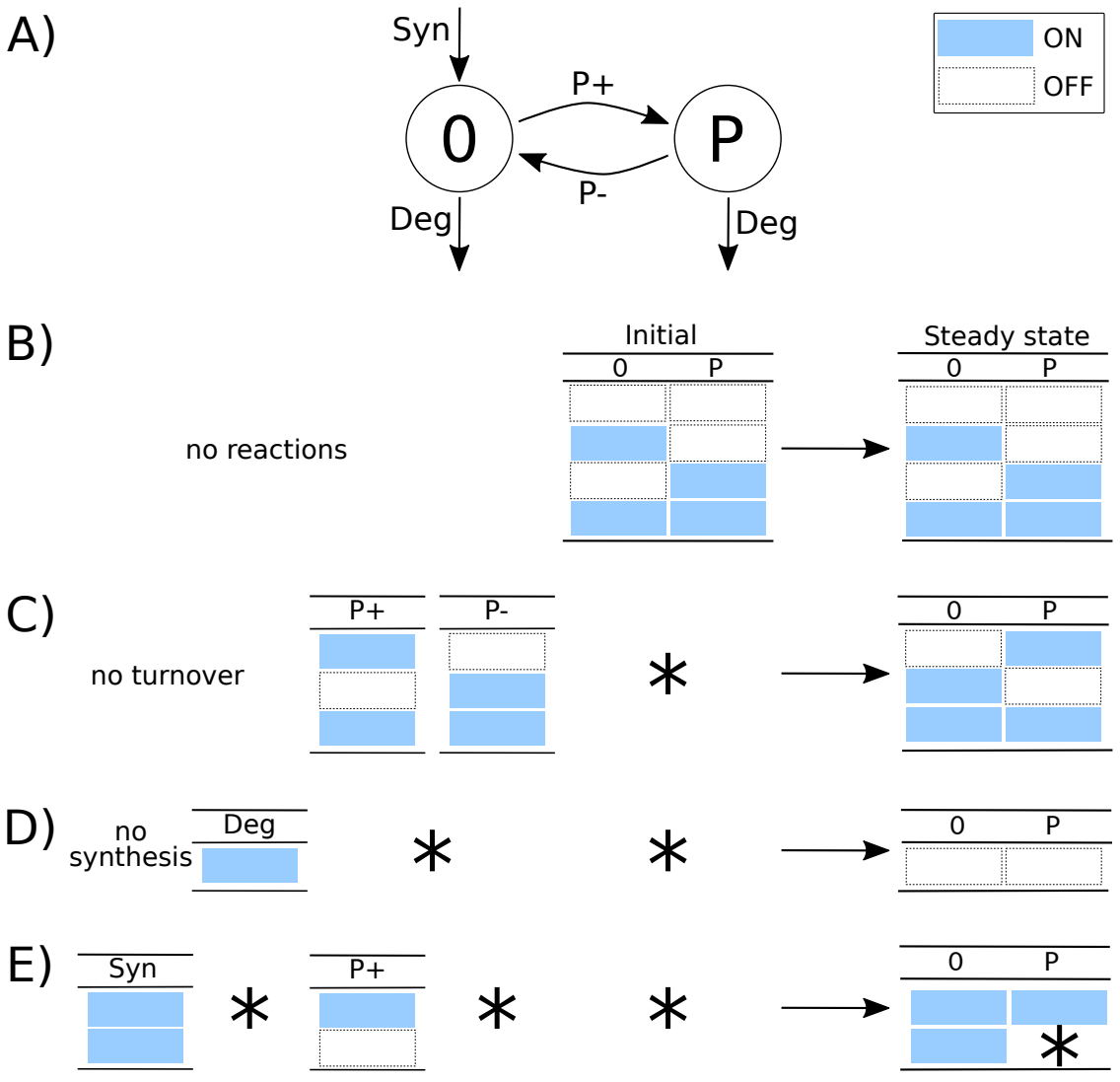

A)
Syn
0
P+
P
P-
Deg
Deg
ON
OFF
B)
Initial
0
P
Steady state
0
P
no reactions
C)
P+
P-
0
P
no turnover
D)
no synthesis
Deg
0
P
E)
Syn
P+
0
P

Figure 1

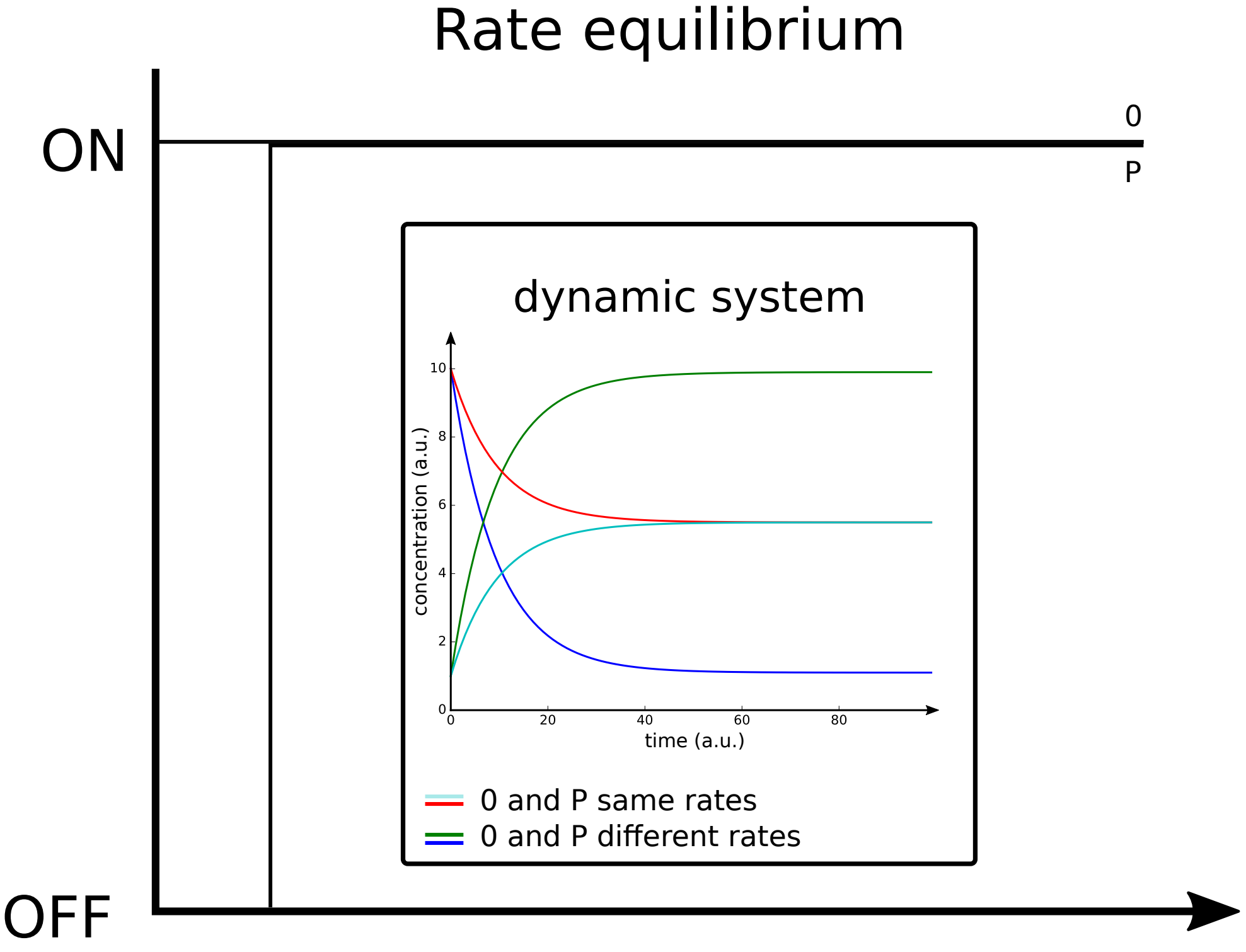

Rate equilibrium
ON
OFF
0
P
dynamic system
concentration (a.u.)
time (a.u.)
0
2
4
6
8
10
0
20
40
60
80
0 and P same rates
0 and P different rates

Figure 2

A) $R^{x}_{(t+1)} = S^{conts}$

$S^{x}_{(t+1)} = (R^{prod} \cap S^{source}) \cup (S^{x} \cap \neg(R^{cons} \cap S^{source})$

B) $R^{x}_{(t+1)} = C \cap S^{conts}$

$S^{x}_{(t+1)} = R^{syn} \cup (C \cap \neg R^{deg} \cap ((R^{prod} \cap S^{source}) \cup (S^{x} \cap \neg(R^{cons} \cap S^{source}))))$

Time & Molecule Smoothing

$$\underbrace{S^{source} \cup \underbrace{(R^{prod} \cap S^{source}}_{\neq})}_{\text{availability of } S^{source}} \cap (R^{syn} \cup \neg R^{deg}))$$

Abbreviation
R^{x}:reaction target
S^{x}:state target

Boolean expression
$\cup$: OR
$\cap$: AND
$\neg$: NOT

Targets
S^{cont} : contingency state target
S^{source} : source state target
R^{syn} : synthesis reaction target
R^{deg} : degradation reactoin target
R^{prod}: production reaction target
R^{cons}: consumption reaction target

C: component $\overbrace{(S_1 \cup \cdots \cup S_n)}^{\text{cannot coexist on single molecule level}} \cap \underbrace{(S_{n+1} \cup \cdots \cup S_m)}_{\text{can coexist on single molecule level}}$

Figure 3

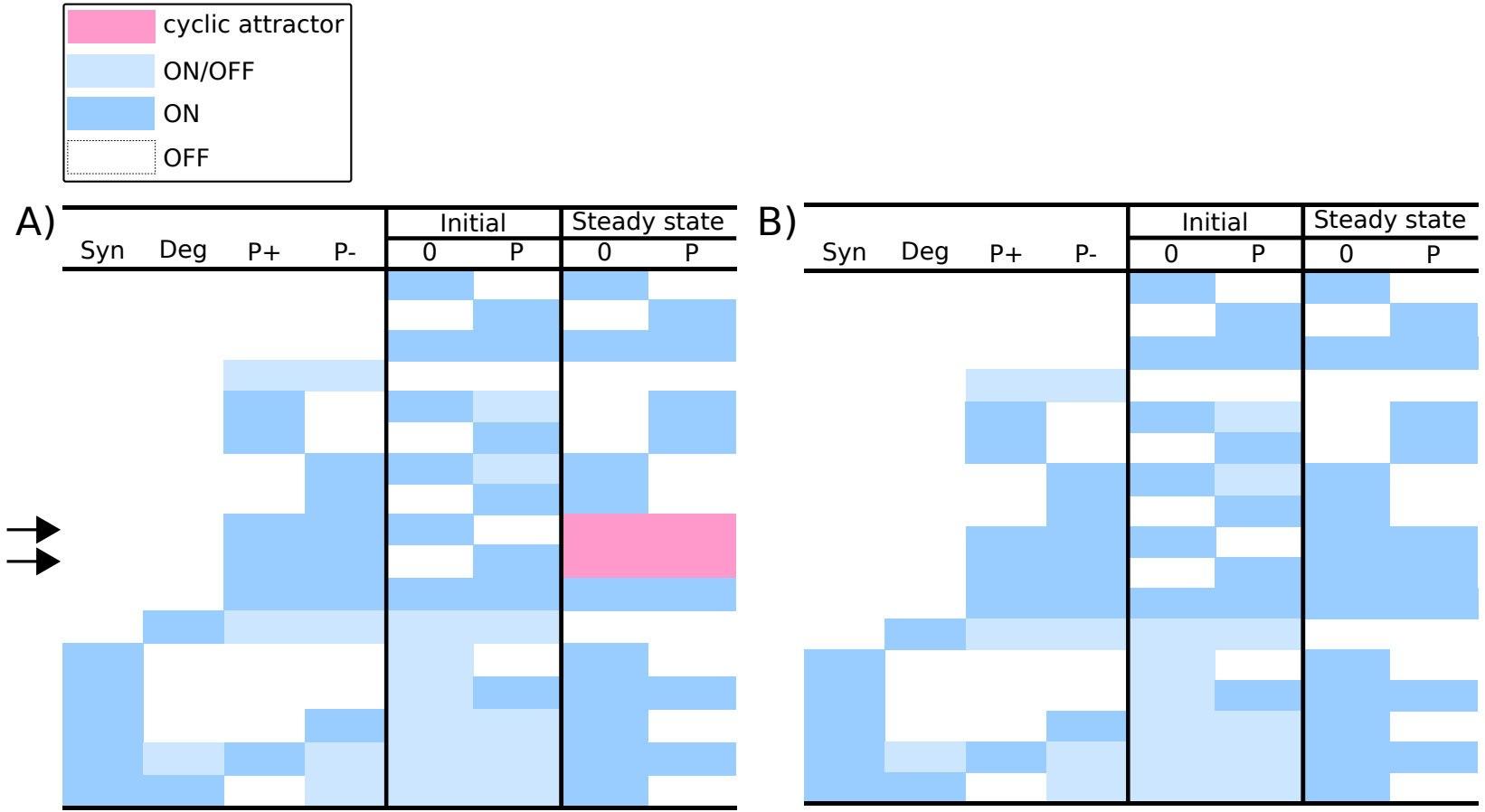
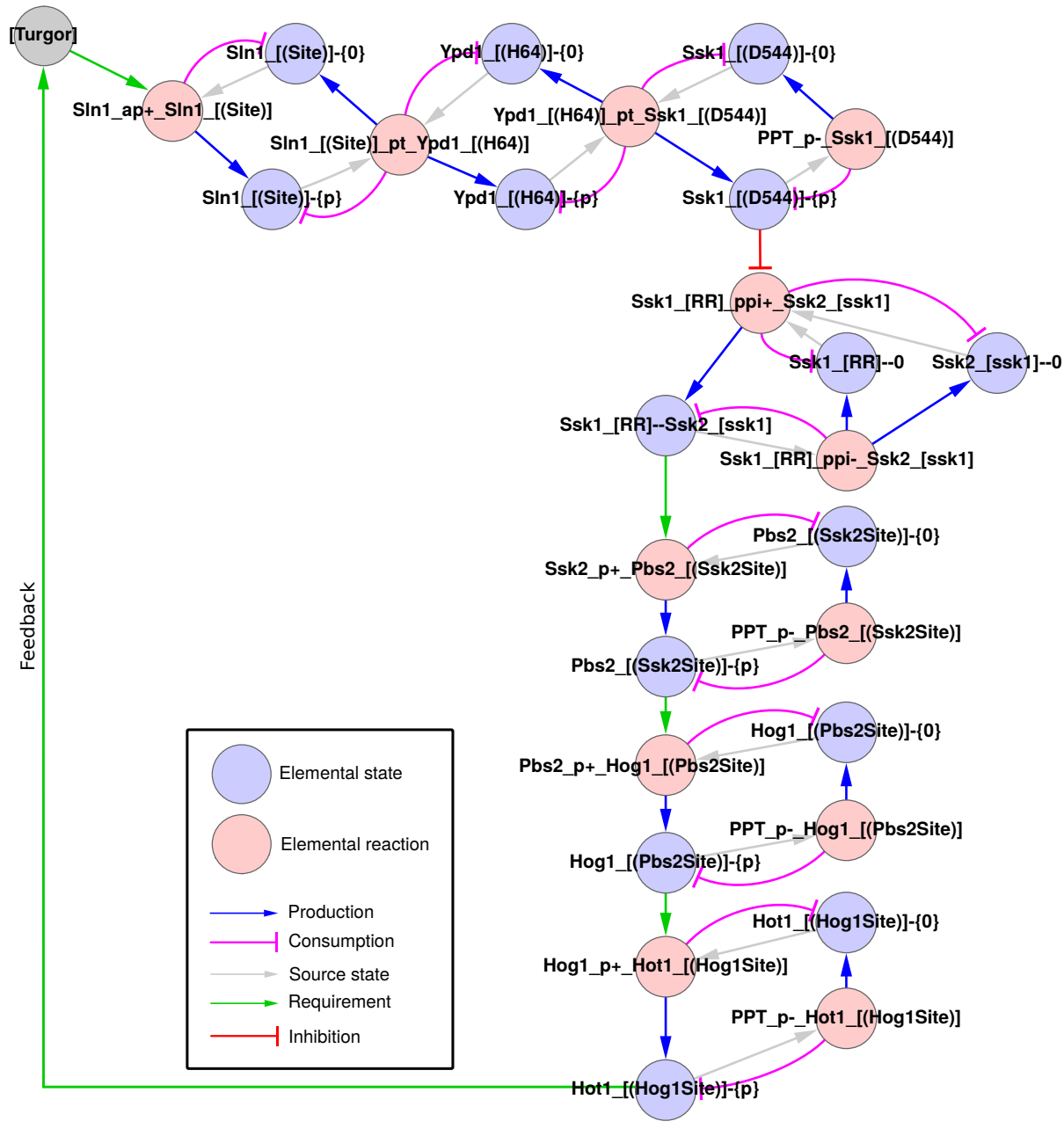

cyclic attractor
ON/OFF
ON
OFF
A)
Syn
Deg
P+
P-
Initial
0
P
Steady state
0
P
B)
Syn
Deg
P+
P-
Initial
0
P
Steady state
0
P

Figure 4

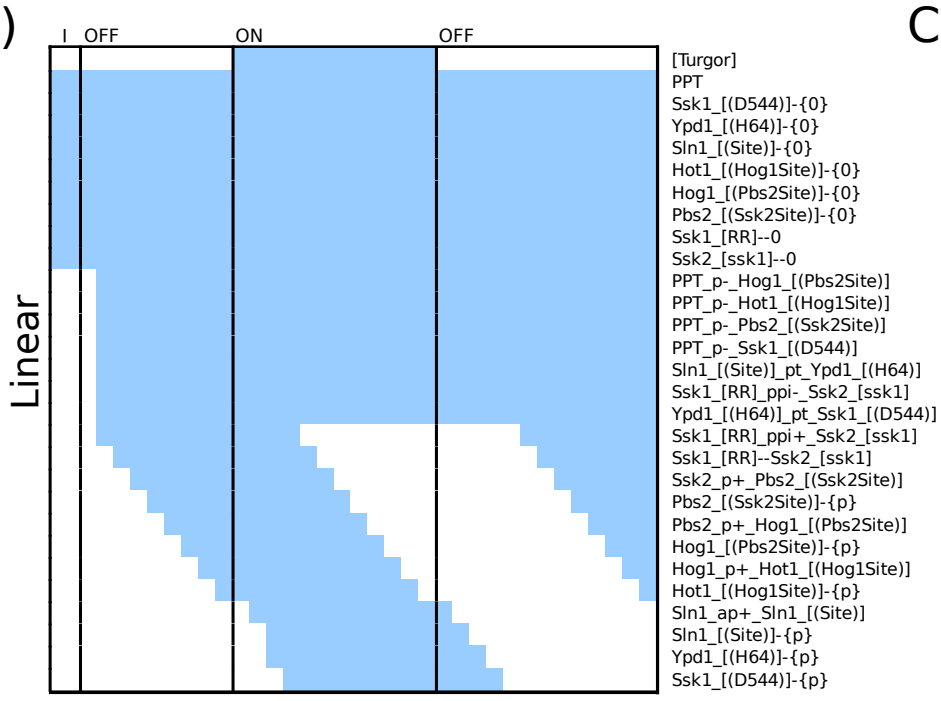
A)



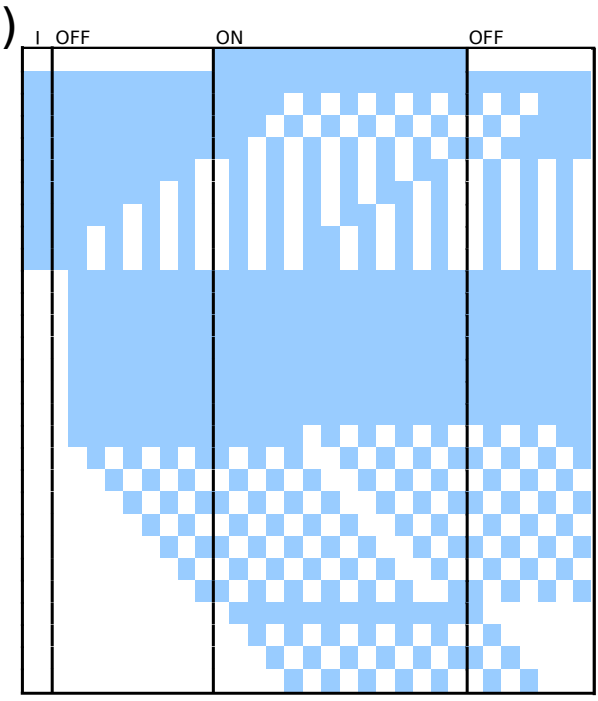
Smoothing

No smoothing

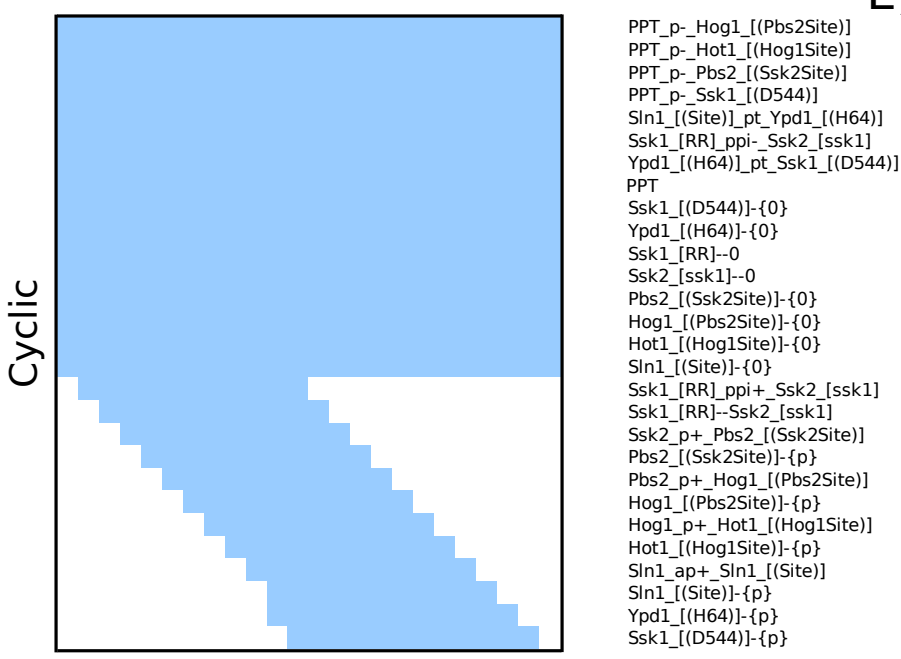
B)



C)



D)



E)

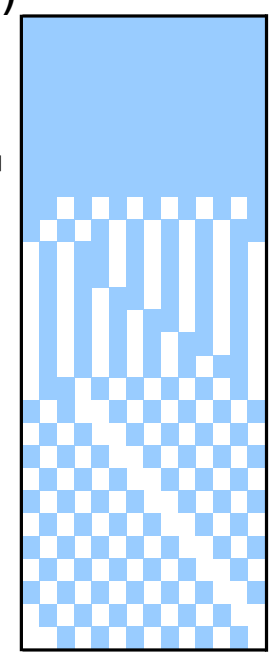


Figure 5

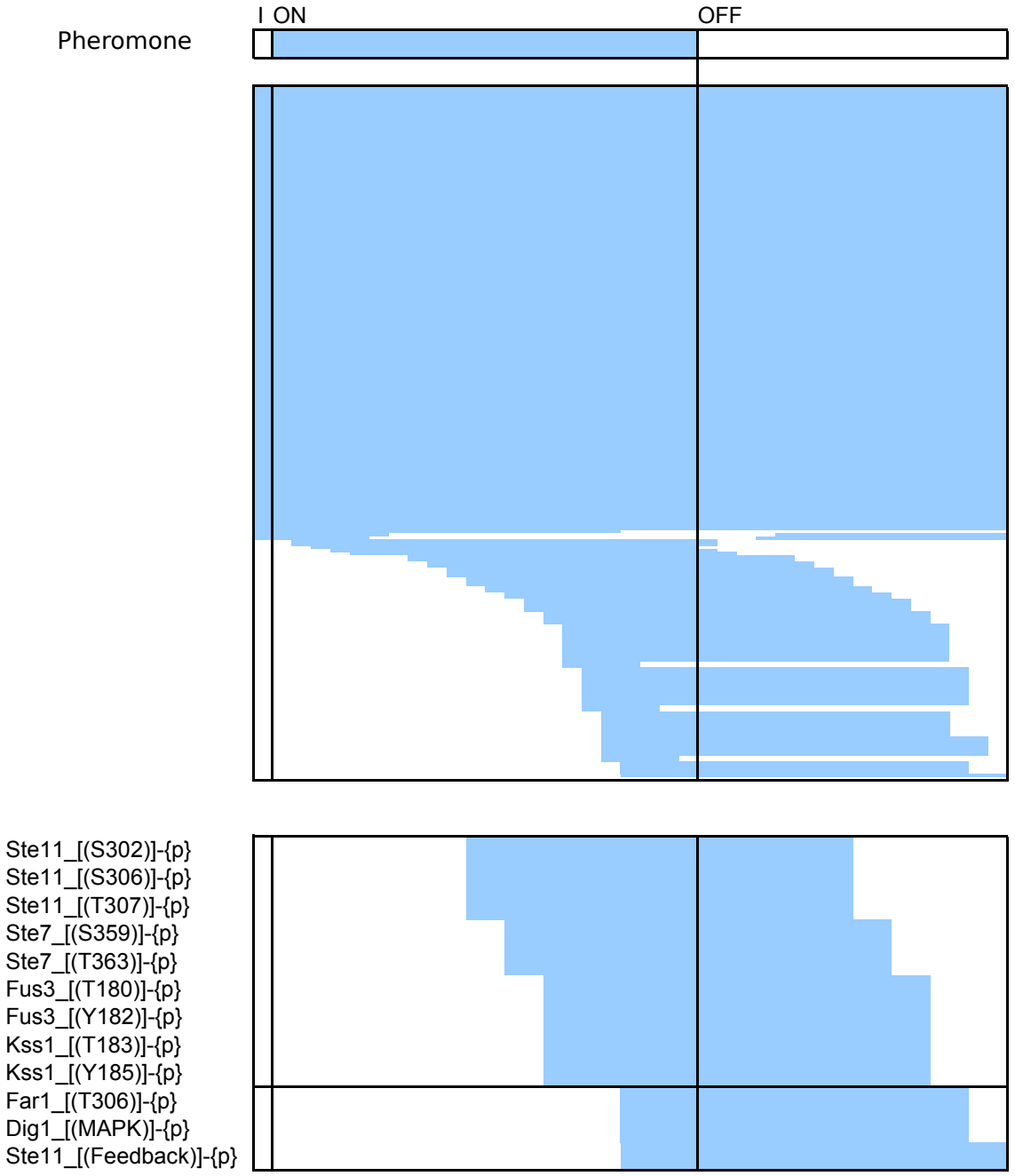

I ON
OFF
Pheromone
Ste11_[(S302)]-{p}
Ste11_[(S306)]-{p}
Ste11_[(T307)]-{p}
Ste7_[(S359)]-{p}
Ste7_[(T363)]-{p}
Fus3_[(T180)]-{p}
Fus3_[(Y182)]-{p}
Kss1_[(T183)]-{p}
Kss1_[(Y185)]-{p}
Far1_[(T306)]-{p}
Dig1_[(MAPK)]-{p}
Ste11_[(Feedback)]-{p}

Figure 6

$$R^{x}_{(t+1)} = \bigcap^{n^{c}} C \cap S^{cont}$$

$$S^{x}_{(t+1)} = \bigcup^{n^{R^{syn}}} R^{syn} \cup \left(\bigcap^{n^{c}} C \cap \bigcap^{n^{R^{deg}}} \neg R^{deg} \cap \left(\bigcup^{n^{R^{prod}}} \left(R^{prod} \cap \bigcap^{n^{S^{source}}} S^{source} \right) \cup \left(S^{x} \cap \bigcap^{n^{R^{cons}}} \neg \left(R^{cons} \cap \bigcap^{n^{S^{source}}} S^{source} \right) \right) \right) \right)$$

Time & Molecule Smoothing

$$S^{source} \cup \underbrace{\left(\bigcup^{n^{R^{prod}}} \underbrace{\left(R^{prod} \cap \bigcap^{n^{S^{source}}} S^{source} \right)}_{\neq} \right)}_{\text{availability of } S^{source}} \cap \left(\bigcup^{n^{R^{syn}}} R^{syn} \cup \bigcap^{n^{R^{deg}}} \neg R^{deg} \right)$$

Abbreviation	Boolean expression	Targets
R^{x}: reaction target	$\cup$: OR	S^{cont} : contingency state target
S^{x}: state target	$\cap$: AND	S^{source} : source state target
n^{x}: number of targets of type x	$\neg$: NOT	R^{syn} : synthesis reaction target
x: target type		R^{deg} : degradation reactoin target
		R^{prod}: production reaction target
		R^{cons}: consumption reaction target

C: component $\overbrace{(S_1 \cup \cdots \cup S_n)}^{\text{cannot coexist on single molecule level}} \cap \overbrace{(S_{n+1} \cup \cdots \cup S_m)}^{\text{can coexist on single molecule level}}$

Figure S1

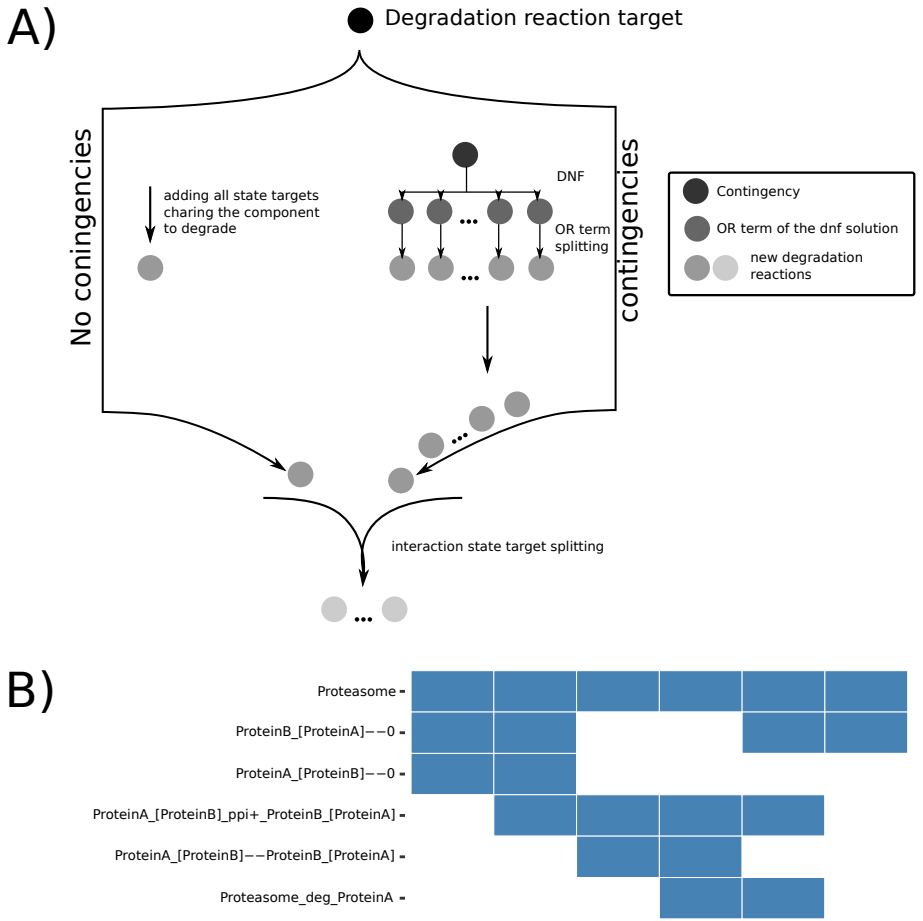

A)
Degradation reaction target
No coningencies
contingencies
adding all state targets
charing the component
to degrade
DNF
OR term
splitting
Contingency
OR term of the dnf solution
new degradation
reactions
interaction state target splitting
B)
Proteasome
ProteinB_[ProteinA]−−0
ProteinA_[ProteinB]−−0
ProteinA_[ProteinB]_ppi+_ProteinB_[ProteinA]
ProteinA_[ProteinB]−−ProteinB_[ProteinA]
Proteasome_deg_ProteinA

Figure S2