A scalable method for parameter-free simulation and validation of mechanistic cellular signal transduction network models.

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

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

* the authors contributed equally

# To whom correspondence should be addressed: mktz75@gmail.com

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. In contrast, current established methods do not support genome-scale mechanistic models of signal transduction networks. These networks encode information through internal states, and dealing with these states leads to scalability issues both in model formulation and execution. While rule based modelling can be used for efficient model definition (through rules) and simulation (through agent based execution), these quantitative models require parametrisation. This introduces yet another layer of uncertainty, due to the sparsity of reliably measured parameters. Hence, parameter-free simulation and validation will be important to support large-scale reconstruction and analysis of signal transduction network models. Here, we present a scalable method for parameter-free simulation of mechanistic signal transduction network models. It is based on rxncon, the reaction-contingency language, which describes the signalling network in terms of elemental reactions and states. We develop two generic update rules for states and reactions, based on detail analysis of two minimal reaction motifs, that can be used to map an arbitrary rxncon network on fully defined bipartite Boolean model. Locally defined update rules are assembled into a functional model without system level optimisation, making the methods suitable for network validation. Furthermore, an underlying model defined solely in terms of molecular reactions and causalities can be used to explain and predict system level behaviour. Taken together, we present a method for parameter-free simulation of mechanistic signal transduction models. Through scalable model definition and simulation, and the independence of quantitative parameters, it opens up for simulation and validation of mechanistic genome-scale models of signal transduction networks.

Introduction

Systems biology aims at the integrative analysis of large-scale biological systems up to whole cells. To realise this goal, we 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. 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 supported by rich toolboxes for network reconstruction, validation and simulation [4], and it constitutes the paradigm for bottom-up modelling. However, these tools cannot be used for signal transduction networks, due to the difference between mass and information transfer networks [5].

Mechanistic modelling of signal transduction is challenging at several levels. First, at the level of model definition: Empirical data on site-specific modifications of, or bonds between, signalling components combine combinatorially into large number of possible configurations, or microstates [6]. To solve this, the community developed simulation tools with adaptive resolution, such as the rule-based modelling languages BioNetGen and Kappa [7,8]. Second, simulation may be prohibitively expensive even with an efficient model definition, as is the case for classical rule-based modelling, in which the full network of microstates must be generated. This was solved by the development of the network-free simulation tool for rule based models, NFsim [9]. Third, quantitative dynamic models require rate laws that must be parametrised in term of rate constants and initial molecule amounts. Reliable information on these quantities is sparse, precluding meaningful parametrisation of most mechanistic models. While the lack of quantitative knowledge is an experimental rather than theoretical challenge (more dedicated biochemistry is needed), a simulation method that voids the need for parametrisation would be extremely helpful in evaluating mechanistic large-scale signalling models until this knowledge gap can be filled.

Parameter-free simulation of cellular networks is typically performed through constraint-based or Boolean methods [10]. In the context of signalling, only the second set of methods is valid: Mass conservation is not informative as the principal entity flowing through the network is information, and hence the basic assumptions of these methods are invalid. In contrast, Boolean models are suited to model information transfer: The network is encoded in nodes and edges, where nodes can be either TRUE or FALSE, and edges define how the node states at time t+1 depend on the node states at time t. Boolean networks have been used extensively in modelling, and different methods and toolboxes have been developed [11]. However, most of these models have not been mechanistic – i.e. they describe functionality at the component level, not at the level of states, which is insufficient to describe the often context dependent activity of signal transduction components. To our knowledge, only three methods have been developed for mechanistic Boolean simulation of signal transduction networks: One derived from SBGN-PD diagrams, one from rule-based models, and one bipartite Boolean modelling (bBM) method based on the rxncon language [12-14]. These can all be used for detailed description of signalling events. However, the first is based on microstate description, inheriting the scalability issues of these approaches [12]; the second requires a full parametrised rule-based model, inheriting the problem of parametrisation [13]; and the third inherited shortcomings in expressiveness and precision from the first generation of the rxncon language [14]. However, the bBM method was successfully used for validation and gap-filling of large signal transduction models [14,15]. Hence, this method – if significantly improved – holds the promise to allow both automatic validation of signal transduction networks and scalable, qualitative simulation of these networks.

The bBM method is based on rxncon, the reaction-contingency language. The rxncon language is tailored for formalising signal transduction models based on empirical data [16]. With rxncon, the reaction network is defined in terms of *elemental states*, i.e. modifications (or lack thereof) at specific residues and bonds (or lack thereof) at specific domains. The network definition is bipartite: *Elemental reactions* are decontextualised reaction events that define how elemental states are *synthesised, degraded, produced* or *consumed*, and *contingencies* define how elemental reactions depend on (combinations of) elemental states that are not changed by the reaction. These contingencies are Boolean combinations of states, and fall into two groups: strict contingencies describe which states are either required for the reaction to occur, or which inhibit it. Furthermore, quantitative contingencies describe a positive or negative effect on the reaction rate. Hence, rxncon supports model definition very close to *experiment*. Models can be made as complex as empirical data requires, but need not be more complex. We recently released rxncon 2.0, addressing previous shortcomings in expressiveness and precision, and providing a formal syntax and semantics [16]. Together, these improvements greatly facilitate the conversion of a rxncon network into mathematical models, and lay the foundation for an improved bBM logic.

Here, we present the first formal semantic mapping of a rxncon network onto a unique bBM with fully specified truth tables. Furthermore, we show that the resulting models meaningfully reproduce the behaviour of signalling processes. We develop two generic update rules for *elemental states* and *elemental reactions* based on the behaviour of two minimal reaction motifs, and demonstrate that these motifs can be connected, LEGO-brick style, into large scale models without any need for optimisation at the system level. Instead, the discrepancy between model behaviour and system levels expectations can be used to identify gaps in the network reconstruction, and hence to improve the model. The method inherits the scalability and precision of the rxncon language, and only a handful of additional assumptions are needed to convert this molecular level knowledge base into a system level model. The implementation is only superficially related to the previous incarnation, and the improved syntax and semantics provides for precise translation of any rxncon network. Taken together, we present a scalable method for parameter-free validation of mechanistic signal transduction network models, taking an important step to close the gap in capabilities between metabolic and signal transduction modelling. Furthermore, the method provides a scalable approach to simulation of signal transduction networks, holding the promise to support modelling at the genome scale.

Results

Generic update rules: basics

The central result presented in this work is a method to derive a parameter-free Boolean model from a rxncon network, in a completely mechanistic fashion.

A Boolean model exists of *nodes* or *targets* that carry a value - true or false - at a given point in time*,* and an update rule describing its value at the next time, formulated as a Boolean function of every node at the current time. The Boolean model we construct is *bipartite:* this means that there exist two classes of nodes, one class for reactions and the other for states. The interpretation we assign to the value of these nodes is as follows.

If a *reaction node* is true, the cellular regulatory network is, at that point in time, in a configuration where it can accommodate that reaction. In the absence of its source state(s), a reaction will not "fire" even though its value is true, as the reaction nodes are purely a description of the regulatory layer of the biological cell. A *state node* is true if there are a sufficient number of molecules carrying that state present in the cell for it to be considered functionally relevant. Both nodes describe *system-level* properties, not molecular ones. A consequence of this is, for example, that two states that are mutually exclusive on a single molecule (*e.g.* a single residue phosphorylated and unphosphorylated), can be simultaneously true in the Boolean system.

Reactions can act on states in four different ways: production, consumption, synthesis and degradation. The behaviour stems directly from the *skeleton rule* underlying the reaction [16]. This rule is similar to a rule-based model rule, such as can be defined in BNGL, but consists solely of a center, without context. We repeat the definitions in that work here, where RHS and LHS refer to the right-hand respectively left-hand side of the skeleton rule. *Production:* a state is produced by a reaction if it appears on the RHS, not on the LHS, but the component carrying the state does appear on the LHS. *Consumption:* a state is consumed by a reaction if it appears on the LHS, not on the RHS, but the component carrying the state does appear in the RHS. *Synthesis:* a state is synthesised by a reaction if it appears on the RHS, and the component carrying the state does not appear on the LHS. *Degradation:* a state is degraded by a reaction if the component carrying the state appears on the LHS, no state mutually exclusive with it appears on the LHS, and the component carrying the state does not appear on the RHS.

In what follows, we study the smallest irreducible motifs containing two states (for modification reactions; Fig 1) or three states (for interactions; Fig S1), and the reactions acting upon them. We define, given an initial configuration for the states and the absence / presence of each of the reactions, the desired steady-state behaviour. The crux of the matter then becomes to find update rules for the states that reproduce this steady state. Our desired behaviour originates in experiment, and is based on the observation that there exists a natural hierarchy between the reaction types introduced in the previous paragraph: *synthesis* is stronger than *degradation* is stronger than *production* is stronger than *consumption*.[1]

[1] Due to the crude concept of Boolean time, we consider a quasi-steady state at each update step. It is then easy to see that a protein that is both synthesised and degraded must be present in the system. In the case where the synthesis reaction is too weak to maintain functional level of the protein, it would be considered off. Similar, production, from the perspective of a specific state, will

Fixing notation: reactions, states, components and all that

A rxncon system will contain N_R reactions denoted by R_i, N_s elemental states denoted by S_i, and N_c components denoted by C_i. For the components that appear without any internal states, such as e.g. those that solely appear as kinases, the component is its own state (which does not appear in the original rxncon system). For those components that do carry internal states, the component can be expanded as a Boolean expression of elemental states grouped by the site (domain or residue) on which they live:

$$C_i = \bigcap_{\substack{site\ j\ on \\ component\ i}} \bigcup_{\substack{state\ k\ on \\ site\ j}} S_k$$

The origin of this expression is in the mutual exclusivity of states that live on the same residue or domain: for each of these sites, at least one of the states living on the site needs to be present for the component itself to be present. A consequence of this is that for components that carry internal states, the dynamic behaviour of the component is fully determined by the states which the component carries.

Furthermore, we define functions mapping states and reactions to Boolean expressions of states. First, the functions $K(R_i)$ list the components which are reacting in reaction R_i:

$$K(R_i) = \bigcap_{\substack{component\ j \\ reacts\ in\ R_i}} C_j$$

Whereas the functions $K(S_i)$ list the components carrying the state S_i (bond states are carried by two components, whereas modifications are carried by one):

$$K(S_i) = \bigcap_{\substack{component\ j \\ carries\ S_i}} C_j$$

In the update rule for the states, we will require the following combinations. First, a reaction together with its source states:

$$R'_i = R_i \bigcap_{\substack{S_j\ consumed \\ by\ R_i}} S_j$$

The neutral state (unmodified, unbound) counterpart for a particular state S_i is denoted by $N(S_i)$.

These notations enable us to write down the synthesis term Σ, which describes whether a component is either directly or indirectly being synthesized:

be dominant over consumption: In the presence of a phosphorylation cycle with both kinases and phosphatases active, both forms will be present. Finally, degradation is dominant over production, as depletion of a protein will deplete the phosphorylated form regardless of kinase activity (except when protected from degradation, as covered in the contingencies below).

$$\Sigma(S_i) = \begin{cases} \displaystyle\bigcup_{\substack{R_j\, synthesizes \\ S_i}} R'_j \text{ for neutral states } S_i \\ \displaystyle\bigcup_{\substack{R_j\, synthesizes \\ N(S_i)}} R'_j \bigcap \bigcup_{\substack{R_k\, produces \\ S_i}} R'_k \text{ for non-neutral states } S_i \end{cases}$$

All systems we considered contain only states “one step” removed from the neutral states, so the expression for non-neutral states describes an active path coming from the synthesized state to the state under consideration. For states that are multiple steps removed from the synthesized neutral state, this expression has to be appropriately amended.

Finally, we denote the Boolean expression representing the contingency for reaction R_i by $L(R_i)$:

$$L(R_i) = \bigcap_j L^!_j(R_i) \bigcap_k \overline{L^\times_k(R_\iota)}$$

where the $L^!_j(R_i)$ and $L^x_k(R_i)$ enumerate the required respectively inhibitory contingencies for reaction R_i, which are themselves possibly nested Boolean expressions.

The expected behaviour of a small reaction circuit and update rule ansatz

The reaction update rules are quite straightforward. We require the strict contingencies[2] for the reaction to be satisfied and the presence of the components on which the reaction acts.

$$R_i(t+1) = K(R_i;t) \bigcap L(R_i;t)$$

The state update rules are more complex and require explicitly the hierarchy between types of reactions that was alluded to above. First of all, if a state is synthesised by any reaction, it will be true. If synthesis is false, the necessary, but not sufficient requirement for the state to be true is that degradation is also false, and that the component(s) carrying the state are present. Now, there are two options for the state to be true: either the state is being produced by some reaction, in which case it is immaterial what the previous value of the state was, or the state was already true and it is not being consumed by any reaction (Fig 1, Fig S1).

$$S_i(t+1) \\ = \Sigma(t) \bigcup \left(K(S_i;t) \bigcap_{\substack{R_k degrades \\ S_i}} \overline{R'_k}(t) \bigcap \left\{ \bigcup_{\substack{R_l\, produces \\ S_i}} R'_l(t) \bigcup \left[S_i(t) \bigcap_{\substack{R_m consumes \\ S_i}} \overline{R'_m}(t) \right] \right\} \right)$$

As can be seen in this formula, the translation of “the state is being produced” *et cetera* contains the primed reactions, meaning the reaction producing that state and the source state(s) of that reaction.

[2] Quantitative contingencies are special: in a Boolean system the reaction rate cannot be increased or decreased: in our analyses we choose to ignore them.

This is due to the semantics of the reaction nodes, that only tell us about the regulatory state of the network.

Testing the generic update rules

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 modification circuitry above (Fig 1A). The models were simulated using BoolNet [17], as described in the methods. The attractor states are visualised in Figure 1C. 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. 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 observe the same phenomenon in the interaction motif, when the reaction cycle is active and at least one component is initiated in and remains in a single form (Fig 1C). We consider these spurious oscillations undesirable in our systems-level description, but note that they would be appropriate for models of single molecules. Nevertheless, the outcome is highly encouraging, as 62 out of 64 models matched the expected behaviour.

Source state smoothing eliminates the spurious oscillations

To eliminate the oscillations that plagued our initial ansatz, we adapted the state target updates by widening the window in which we checked for source state availability: A reaction needs the source state to be present, or to be produced. To achieve this, in formula (8) we substitute for the reactions producing the state, *R'_l* the following:

$$R''_l(t) = R_l(t) \bigcap_{\substack{S_j\ consumed \\ by\ R_l}} S_j(t) \bigcup S_j(t+1)$$

Where the Sj(t+1) is the full expression (8), without this substitution.

We consider this adaptation quite natural: there are large numbers of molecules undergoing the same set of reactions. Therefore, it is highly unlikely that all of these reactions are temporally completely in phase, justifying a smoothing over molecules. In addition, the time scale in a Boolean model is basically set by the slowest of the reactions, since all rate constants are absent. For molecule pools acted upon by reactions that are faster than the slowest in the system, it is likely that they will pass through mutually exclusive states within the window of a Boolean time step, justifying a “time smoothing”. Both effects are captured by the reworked update rules.

The smoothing assumption only breaks down in the context of few molecules and low reaction rates, and there are cases in which smoothing is inappropriate. In other work [18], we have introduced a hybrid model containing both the molecular reactions and states described here, and additionally macroscopic reactions and states that are governed by the "un-smoothed" update rules. However, for most states in a signal transduction network, the smoothed update rule is more appropriate. 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 1D, Fig S1D). The oscillatory behaviour disappeared, but no other simulation results changed. Hence, the simulation results exactly match the behaviour we expect from a model of these reaction motifs.

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 2; 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. We deemed this too many for an exhaustive search, and decided to use a generic start state for all simulations: All neutral elemental states (that is, unbound binding domains and neutral modifications) 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 2B). 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 2B, 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 S2A), where we see the signal passing the network despite spurious 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 (Fig 2B): 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 regulated 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 the molecular level knowledge in 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 [19]. To simulate this, we linked the most downstream component to the input that turns the pathway off (dashed line in Fig 2A). We repeated the model creation and simulation, using the initial steady state of the linear model as starting condition. As shown in Figure 2C, 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 S2B). Here, the pathway signal is completely washed out by the spurious oscillations and breaks down to a two-state cyclic attractor. 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 mechanistically detailed rule based model (RBM) [20]. The original RBM contains 229 rules with 200 parameters (166 unknown) that define how 18 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 [21]. 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 translation of the RBM, which is described elsewhere [16], is defined by 95 elemental reactions and 231 (non-zero) contingencies. We generated the bBM using the smoothed update rules, which produced a bipartite Boolean model with 130 reaction targets (due to separation of bidirectional reactions and duplication of degradation reaction targets) and 118 state targets. With 248 state variables, the model is too large to use an exhaustive search of initial states (statespace = 2^{248}; ca 10^{74} 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. We found that the pathway was constitutively active and unresponsive to pheromone. First, we examined if this was due to the interpretation of quantitative effects that are lost in the Boolean model. However, neither ignoring all nor including all K+/K- contingencies solves the problem. Furthermore, the original RBM was never simulated and proven functional. Hence, we proceeded with the minimal model (ignoring quantitative contingencies) and looked deeper into the pathway behaviour, finding that it activates in the absence of signal due to constitutive release of Ste4, which represents the beta/gamma subunit of the trimeric G-protein at the top of the cascade. To address the problem, we change two (out of 87) quantitative contingencies into qualitative contingencies, indicating that a much simpler model would suffice to capture the key features of the pathway. In addition, we needed to limit turnover of Ste4 bound Gpa1 (to prevent signal-independent release of unbound Ste4, the activator of the pathway) and to remove Fus3 dependent degradation of Ste12 which made the pathway “single-shot”. The final model with changes can be found in Table S2, and the simulation trajectories are shown in figure 3. This updated version of the model responds to pheromone exposure and withdrawal as expected.

Discussion

Here, we present a qualitative simulation method for large-scale mechanistically detailed signal transduction network models. The formalism is based on Boolean logic and can be simulated and studied by a standard package such as BoolNet. However, we present a fundamentally new model generation and simulation concept. First, we create a bipartite model at the level of elemental reactions and contingencies, capturing the key elements in signal transduction at an appropriate resolution for mechanistic modelling. Second, based on detailed analysis of two minimal reaction motifs, 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 from molecular mechanisms, without optimisation at the system level.

The unique mapping from rxncon to an executable bBM that predicts system behaviour is highly non-trivial. 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 set truth tables, and that this Boolean model meaningfully predicts system level behaviour. Thus, the bBM logic we present here bridges the microscopic (biochemical reactions) and macroscopic (input-output) levels of cellular signal transduction, fulfilling the requirements for the cellular “mechanics” proposed by Hlavacek and Faeder - at least qualitatively [6].

This has far-reaching implications: First, it provides an efficient validation tool in the model building process. In the context of rule-based modelling, it allows the model construction to be separated in to two phases: A qualitative and a quantitative phase. Boolean models are computationally inexpensive, and the automatic model generation supports iterative model creation, analysis and improvement. In addition, we are better equipped with knowledge at the qualitative level, suggesting that this level should be optimised first. As rxncon supports compilation into both RBMs and bBMs (as well as several graphical formats), it can be used to facilitate this process [22]: The structural model can be created and validated using graphical tools and bBM simulation, and later the improved network can be used to create a rule-based model. Hence, the more expensive parameterisation cycles can be performed after the qualitative model has passed the validation process. Second, the bBM method can be used for validation of large-scale signal transduction networks. Previously, large scale reconstruction of signal transduction has been limited to graphical maps that cannot be executed [23-25]. The tool we present here changes this: We can now validate – through simulation – large-scale reconstructions of signal transduction networks in the rxncon language.

The bipartite Boolean modelling formalism we presented here if fundamentally different from its previous incarnation [14]. First, it has been designed to capture system level behaviour, not the states of individual molecule instances in the network, meaning that states that would be mutually exclusive on a single molecule can be true at the same time. As we show in figure S2, this is critical for meaningful system level predictions. Second, we used a constructive approach, defining all possible behaviours at the level of two families of minimal reaction motifs, to design two generic update rules. These update rules can be used to map any rxncon system, even extended by new reaction types, through the interpretation of the flexible skeleton rule definition as synthesis, degradation production, or consumption of different states. Third, the method we present here inherits the expressiveness and flexibility of rxncon 2.0, including the explicit representation of neutral states [16]. Fourth, the reimplementation has improved the model generation, enabled the use of different export options, and improved the model creation and analysis workflow. While the previous incarnation worked well in many instances [14,15,26], these models had issues with certain reaction types (most notably degradation) and spurious oscillations. That the latter appeared so rare was due to the implicit dominance of modified states: Neutral states were not explicitly represented, making modification or binding reactions dominant over reactions that returned components to their neutral state. Here, we eliminate this artificial hierarchy, which we consider undesirable, and make the neutral states explicit. However, most of these states are constantly true in our simulations. This may have two reasons: First, it could reflect biology: There would be a constant pool of unmodified components as long as 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, [27]). If so, the formalism we present here helps us make this information bias explicit, and will allow us to integrate the regulation on these reactions as the knowledge becomes available.

The tools we present here enable a scale-shift in signal transduction modelling. Hitherto, executable signalling models have been mechanistically detailed or large-scale, but not both. Most mechanistic large-scale reconstructions are technically microstate models that could be simulated after parametrisation. However, they are actually divided over several unconnected modules and could hence not be simulated at the system level [23-25]. Rule-based modelling have been used to build relatively large models [21,28], but even these models are limited to few (18 in these cases) components and parametrisation is already an outstanding challenge. In contrast, comprehensive signalling models will need to account for hundreds or thousands of signalling components, carrying many thousands distinct elemental states. Here, we present a method that can deal with mechanistic signalling networks at this scope. In ongoing work, we are building a comprehensive mechanistic model of the yeast cell division cycle, at residue resolution, currently accounting for 229 proteins, 1248 elemental reactions and 801 contingencies. Using the rxncon language and the parameter-free simulation tool presented here, it is possible to build, simulate and analyse models at this scale (Münzner *et al.*, in preparation; [18]). The method is qualitative, but this may be an advantage given the sparsity of reliable quantitative information on rate constants. In addition, even metabolic modelling – clearly the state-of-the-art in genome-scale modelling – is limited to qualitative or semi-quantitative simulation methods at the genome scale.

Taken together, we present a parameter-free model creation and simulation method that 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).

Appendix: Methods

Model creation and analysis

The creation of the rxncon models are described elsewhere. The High Osmolarity Glycerol (HOG) model was taken from [14] and adapted to rxncon 2.0. The pheromone (PHER) model was translated from the yeastpheromonemodel.org wiki as described in [16]. The bipartite Boolean model files were created with the rxncon compiler software, by calling the "rxncon2boolnet.py" script with default setting. The rxncon software is open source, distributed under the lGPL licence, and can either be downloaded from https://github.com/rxncon/rxncon or installed from the python package index with "pip install rxncon". The rxncon model files are available as Table S1 (HOG model), Table S2 (final PHER model) or through download from https://github.com/rxncon/models/ (initial PHER model; YeastPheromoneModel.xls).

The rxncon2boolnet.py script generates three files. First, the bipartite Boolean model file (<ModelName>.boolnet) contains the update rules using states and reaction IDs. Second, the symbol mapping file (<ModelName>_symbols.csv) defines which IDs correspond to which states and reactions in the rxncon file. Third, the initial vector (<ModelName>_initial_vals.csv') sets the initial state of the Boolean simulation.

Model simulation was done with the R CRAN package BoolNet [17]. To facilitate simulation, we prepared an R script that can be downloaded from https://github.com/rxncon/tools (BoolNetSim.R). To use this script through R studio:

- Save the network files and the R script into a single directory.
- Start RStudio.
- Open a new project and create it in the directory where you saved your files.
- Make sure your model files are located in the project folder.
- Open the R script. Set the filePrefix in the R script to <model>.
- Execute the entire script by selecting all text (ctrl+a) and pressing ctrl+enter.

The script generates five files: (i) <ModelName>.pdf, which graphically displays the simulation trajectory from initial state to the attractor, (ii) <ModelName>_trajectory_first.csv, with the trajectory as values (0/1) in tabular format, (iii) <ModelName>_2.pdf, which graphically displays the simulation trajectory from the attractor (useful to distinguish a point attractor (two columns) from a cyclic attractor (>2 columns), (iv) <ModelName>_trajectory_second.csv, with the second trajectory in tabular format, and (v) <ModelName>_new_attractor.csv, with the new attractor as an initial values file.

Where "<ModelName>" is the file name (without extension) of your rxncon 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.

Figure Legends

Figure 1: Behaviour of a minimal modification motif. (A) The motif includes two states of A, unphosphorylated (A-{0}) and phosphorylated (A-{P}). The motif contains up to four different reactions: Component A can be synthesised (in its neutral state A-{0}), degraded (in either state), phosphorylated (consumes A-{0}, produces A-{P}) or dephosphorylated (consumes A-{P}, produces A-{0}). **(B)** The expected steady state as a function of initial state and active reactions. (I) In the absence of any reactions, the equilibrium state will be identical to the initial state. (II-III) In absence of synthesis or degradation, but in the presence of component A (A-{0} or A-{P} true), the equilibrium depends on the (de)phosphorylation reactions. With only one of these reactions, only the fully (de)phosphorylated form is present. However, with both reactions present we expect both elemental states to be present at steady state. (IV) With degradation but not synthesis active, the protein will be depleted and both states will be false. (V) With active synthesis, the neutral state will always be present. The phosphorylated state will only be present if there is also a phosphorylation reaction or if the state is initially present and both the degradation and dephosphorylation reactions are off. **(C)** The simulation outcome with the update rules in the original ansatz. The results correspond to the expected behaviour for 62 out of 64 configurations. The exception are the cyclic attractors in block (III), where both phosphorylation and dephosphorylation are active and only one of the elemental states are initiated as true. **(D)** The simulation outcome with the smoothed update rules. The results are identical to the expected attractor.

Figure S1: Behaviour of a minimal interaction motif. (A) The motif includes the unbound state of A (A--0), the unbound state of B (B--0) and the bound state (A--B). As in figure 1, the motif contains up to four different reactions: Component A can be synthesised (in its neutral state A--0), degraded (in either state), A and B can bind (ppi+; consumes A--0 and B--0, produces A--B) or dissociate (ppi-consumes A--B, produces A--0 and B--0). Not that degradation of A in the A--B dimer releases B--0, hence this reaction is a degradation reaction of A and conditional production reaction for B--0. **(B)** The expected steady state as a function of initial state and active reactions. As in figure 1, the initial state is preserved when no reaction is active. However, the steady state in the presence of reactions is more complex, as both unbound states are necessary for the reaction to fire - which affects both the generation of the A--B state and the depletion of the unbound states. With degradation and without synthesis, A--0 and A--B is removed, releasing B--0 in the latter case. Finally, synthesis of A only leads to A--B in the presence of the forward reaction, in which case B--0 is depleted unless A--B is turned over by degradation or dissociation. **(C)** The simulation outcome with the update rules in the original ansatz. The results correspond to the expected behaviour, except when a component is present in only one state. This happens in the analogous case to the spurious oscillations in the modification motif in figure 1, but also when A is synthesised if B is only present in one form. **(D)** The simulation outcome with the smoothed update rules. The results are identical to the expected attractor.

Figure 2: From reactions to a functional pathway. We used the smoothed update rules to generate and simulate a model of the Sln1 branch of the high osmolarity glycerol (HOG) pathway. (A) The pathway visualised as a rxncon regulatory graph. In the absence of turgor, Sln1 stays unphosphorylated. As turgor increases, the auto-phosphorylation of Sln1 initiates 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. The dashed line indicates a feedback loop that is included in the cyclic version only. (B) Simulation of a linear version

of the model using source state smoothing of the update rules. (I) We use our default assumptions on the initial state and simulate the model until we reach an attractor (first OFF trajectory). (II) We activate the system by turning [Turgor] ON and simulate again (ON trajectory) until we reach an attractor state. (III) 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) 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 from the initial OFF attractor (see panel B), and note that the system oscillates as expected: The trajectory is cyclic, where the last time step is followed by the first, and turgor is now a model variable that turns on and off during the cycle (grey row in the heatmap).

Figure S2: Smoothing is required to make pathway simulation interpretable. We repeated the simulation of the HOG pathway with the un-smoothed update rules from the initial ansatz. (A) Simulation of the linear model without smoothing. The signal goes through the pathway, but analysis is complicated by the spurious oscillations as the system no longer converges on point attractors. Each of the three simulation trajectories (I-III) ends with a cyclic attractor of length two. (B) Simulation of the cyclic HOG model without smoothing. Here, the entire oscillation cycle breaks down into a period two oscillator involving all the states (and their complements) and reactions that transmit the information.

Figure 3: The updated pheromone model responds as expected to pheromone. We generated a bBM form the updated pheromone model and (I) simulated it in the absence of pheromone (unbound pheromone set to false) until the first steady state, where (II) free pheromone was set to true, representing pheromone stimulation, and the simulation repeated until next steady state was reached, before (III) pheromone was removed (by setting both free and bound pheromone to false) and the model simulated to the next steady state. The pathway turns on and off as expected, and finds a natural off state in the first simulation despite two activation pulses that go through the pathway. These are due to proteins that activate the pathway in their neutral states: The upstream Ste4, as discussed in the text, and the Ste12 transcription factor - which, according to the model, is active when unbound to Dig1 and Dig2. However, both pulses are transient and the steady state is robust against these transient dynamics.

References

1 Fisher, J. & Henzinger, T. A. Executable cell biology. *Nat Biotechnol* **25**, 1239-1249, doi:10.1038/nbt1356 (2007).

2 Herrgard, M. J. *et al.* A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nat Biotechnol* **26**, 1155-1160, doi:10.1038/nbt1492 (2008).

3 Thiele, I. *et al.* A community-driven global reconstruction of human metabolism. *Nat Biotechnol* **31**, 419-425, doi:10.1038/nbt.2488 (2013).

4 Thiele, I. & Palsson, B. O. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature protocols* **5**, 93-121, doi:10.1038/nprot.2009.203 (2010).

5 Münzner, U., Lubitz, T., Klipp, E. & Krantz, M. in *Systems Biology* (eds J. Nielsen & S. Hohmann) 215-242 (Wiley, 2017).

6 Hlavacek, W. S. & Faeder, J. R. The complexity of cell signaling and the need for a new mechanics. *Sci Signal* **2**, pe46, doi:10.1126/scisignal.281pe46 (2009).

7 Blinov, M. L., Faeder, J. R., Goldstein, B. & Hlavacek, W. S. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics* **20**, 3289-3291, doi:10.1093/bioinformatics/bth378 (2004).

8 Danos, V., Feret, J., Fontana, W., Harmer, R. & Krivine, J. in *CONCUR 2007 – Concurrency Theory: 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007. Proceedings* (eds Luís Caires & Vasco T. Vasconcelos) 17-41 (Springer Berlin Heidelberg, 2007).

9 Sneddon, M. W., Faeder, J. R. & Emonet, T. Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nat Methods* **8**, 177-183, doi:10.1038/nmeth.1546 (2011).

10 Machado, D. *et al.* Modeling formalisms in Systems Biology. *AMB Express* **1**, 45, doi:10.1186/2191-0855-1-45 (2011).

11 Abou-Jaoude, W. *et al.* Logical Modeling and Dynamical Analysis of Cellular Networks. *Front Genet* **7**, 94, doi:10.3389/fgene.2016.00094 (2016).

12 Handorf, T. & Klipp, E. Modeling mechanistic biological networks: an advanced Boolean approach. *Bioinformatics* **28**, 557-563, doi:10.1093/bioinformatics/btr697 (2012).

13 Kolczyk, K., Samaga, R., Conzelmann, H., Mirschel, S. & Conradi, C. The Process-Interaction-Model: a common representation of rule-based and logical models allows studying signal transduction on different levels of detail. *BMC bioinformatics* **13**, 251, doi:10.1186/1471-2105-13-251 (2012).

14 Flottmann, M., Krause, F., Klipp, E. & Krantz, M. Reaction-contingency based bipartite Boolean modelling. *BMC Syst Biol* **7**, 58, doi:10.1186/1752-0509-7-58 (2013).

15 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* **1**, 15007, doi:10.1038/npjsba.2015.7

http://www.nature.com/articles/npjsba20157#supplementary-information (2015).

16 Romers, J. C. & Krantz, M. rxncon 2.0: a language for executable molecular systems biology. *bioRxiv*, doi:10.1101/107136 (2017).

17 Mussel, C., Hopfensitz, M. & Kestler, H. A. BoolNet--an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics* **26**, 1378-1380, doi:10.1093/bioinformatics/btq124 (2010).

18 Münzner, U. *From birth to birth. A cell cycle control network of S. cerevisiae* PhD thesis, Humboldt-Universität zu Berlin, (2017).

19 Klipp, E., Nordlander, B., Kruger, R., Gennemark, P. & Hohmann, S. Integrative model of the response of yeast to osmotic shock. *Nat Biotechnol* **23**, 975-982, doi:10.1038/nbt1114 (2005).

20 Thomson, T. M. *Yeast Pheromone Model*, <http://yeastpheromonemodel.org> (

21 Thomson, T. M. *et al.* Scaffold number in yeast signaling system sets tradeoff between system output and dynamic range. *Proc Natl Acad Sci U S A* **108**, 20265-20270, doi:10.1073/pnas.1004042108 (2011).

22 Romers J., Thieme S., Münzner U. & M., K. in *Modeling Biomolecular Site Dynamics Methods in Molecular Biology* (Springer, Submitted (invited)).

23 Fabregat, A. *et al.* The Reactome pathway Knowledgebase. *Nucleic acids research* **44**, D481-487, doi:10.1093/nar/gkv1351 (2016).

24 Kawakami, E. *et al.* Network analyses based on comprehensive molecular interaction maps reveal robust control structures in yeast stress response pathways. *Npj Systems Biology And Applications* **2**, 15018, doi:10.1038/npjsba.2015.18

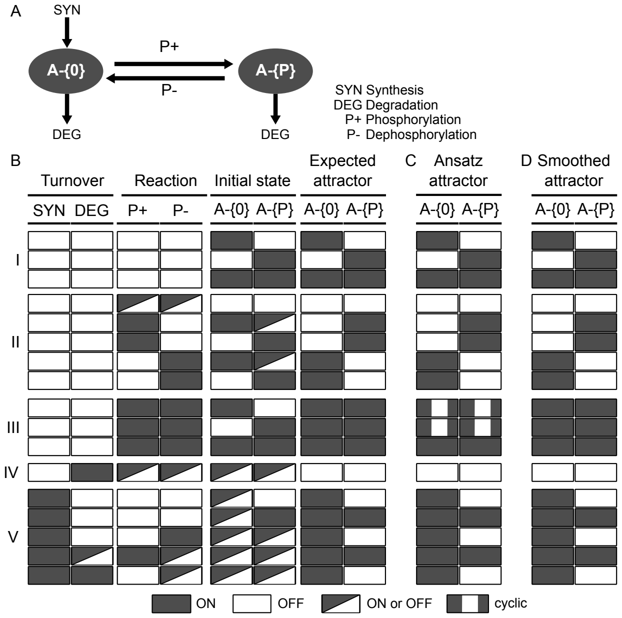
http://www.nature.com/articles/npjsba201518#supplementary-information (2016).

25 Kuperstein, I. *et al.* Atlas of Cancer Signalling Network: a systems biology resource for integrative analysis of cancer data with Google Maps. *Oncogenesis* **4**, e160, doi:10.1038/oncsis.2015.19 (2015).

26 Mori, T., Flottmann, M., Krantz, M., Akutsu, T. & Klipp, E. Stochastic simulation of Boolean rxncon models: towards quantitative analysis of large signaling networks. *BMC Syst Biol* **9**, 45, doi:10.1186/s12918-015-0193-8 (2015).

27 Tiger, C. F. *et al.* A framework for mapping, visualisation and automatic model creation of signal-transduction networks. *Mol Syst Biol* **8**, 578, doi:10.1038/msb.2012.12 (2012).

28 Creamer, M. S. *et al.* Specification, annotation, visualization and simulation of a large rule-based model for ERBB receptor signaling. *BMC Syst Biol* **6**, 107, doi:10.1186/1752-0509-6-107 (2012).



	B Turnover		B Reaction		B Initial state		B Expected attractor		C Ansatz attractor		D Smoothed attractor	
	SYN	DEG	P+	P-	A-{0}	A-{P}	A-{0}	A-{P}	A-{0}	A-{P}	A-{0}	A-{P}
I	□	□	□	□	■	□	■	□	■	□	■	□
	□	□	□	□	□	■	□	■	□	■	□	■
	□	□	□	□	■	■	■	■	■	■	■	■
II	□	□	◪	◪	□	□	□	□	□	□	□	□
	□	□	■	□	■	◪	□	■	□	■	□	■
	□	□	■	□	□	■	□	■	□	■	□	■
	□	□	□	■	■	◪	■	□	■	□	■	□
	□	□	□	■	□	■	■	□	■	□	■	□
III	□	□	■	■	■	□	■	■	◫	◫	■	■
	□	□	■	■	□	■	■	■	◫	◫	■	■
	□	□	■	■	■	■	■	■	■	■	■	■
IV	□	■	◪	◪	◪	◪	□	□	□	□	□	□
V	■	□	□	□	◪	□	■	□	■	□	■	□
	■	□	□	□	◪	■	■	■	■	■	■	■
	■	□	□	■	◪	◪	■	□	■	□	■	□
	■	◪	■	◪	◪	◪	■	■	■	■	■	■
	■	■	□	◪	◪	◪	■	□	■	□	■	□

Figure 1

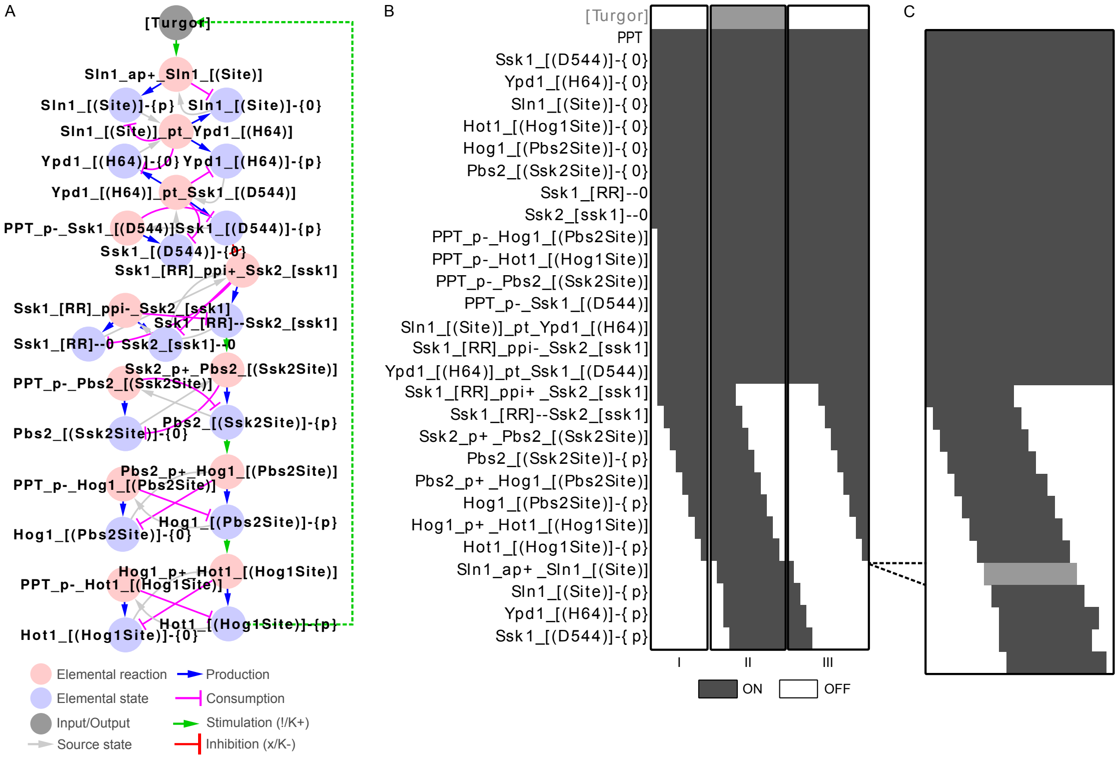

A
[Turgor]
Sln1_ap+_Sln1_[(Site)]
Sln1_[(Site)]-{p}
Sln1_[(Site)]-{0}
Sln1_[(Site)]_pt_Ypd1_[(H64)]
Ypd1_[(H64)]-{0}
Ypd1_[(H64)]-{p}
Ypd1_[(H64)]_pt_Ssk1_[(D544)]
PPT_p-_Ssk1_[(D544)]
Ssk1_[(D544)]-{p}
Ssk1_[(D544)]-{0}
Ssk1_[RR]_ppi+_Ssk2_[ssk1]
Ssk1_[RR]_ppi-_Ssk2_[ssk1]
Ssk1_[RR]--Ssk2_[ssk1]
Ssk1_[RR]--0
Ssk2_[ssk1]--0
Ssk2_p+_Pbs2_[(Ssk2Site)]
PPT_p-_Pbs2_[(Ssk2Site)]
Pbs2_[(Ssk2Site)]-{p}
Pbs2_[(Ssk2Site)]-{0}
Pbs2_p+_Hog1_[(Pbs2Site)]
PPT_p-_Hog1_[(Pbs2Site)]
Hog1_[(Pbs2Site)]-{p}
Hog1_[(Pbs2Site)]-{0}
Hog1_p+_Hot1_[(Hog1Site)]
PPT_p-_Hot1_[(Hog1Site)]
Hot1_[(Hog1Site)]-{p}
Hot1_[(Hog1Site)]-{0}
Elemental reaction
Elemental state
Input/Output
Source state
Production
Consumption
Stimulation (!/K+)
Inhibition (x/K-)
B
[Turgor]
PPT
Ssk1_[(D544)]-{0}
Ypd1_[(H64)]-{0}
Sln1_[(Site)]-{0}
Hot1_[(Hog1Site)]-{0}
Hog1_[(Pbs2Site)]-{0}
Pbs2_[(Ssk2Site)]-{0}
Ssk1_[RR]--0
Ssk2_[ssk1]--0
PPT_p-_Hog1_[(Pbs2Site)]
PPT_p-_Hot1_[(Hog1Site)]
PPT_p-_Pbs2_[(Ssk2Site)]
PPT_p-_Ssk1_[(D544)]
Sln1_[(Site)]_pt_Ypd1_[(H64)]
Ssk1_[RR]_ppi-_Ssk2_[ssk1]
Ypd1_[(H64)]_pt_Ssk1_[(D544)]
Ssk1_[RR]_ppi+_Ssk2_[ssk1]
Ssk1_[RR]--Ssk2_[ssk1]
Ssk2_p+_Pbs2_[(Ssk2Site)]
Pbs2_[(Ssk2Site)]-{p}
Pbs2_p+_Hog1_[(Pbs2Site)]
Hog1_[(Pbs2Site)]-{p}
Hog1_p+_Hot1_[(Hog1Site)]
Hot1_[(Hog1Site)]-{p}
Sln1_ap+_Sln1_[(Site)]
Sln1_[(Site)]-{p}
Ypd1_[(H64)]-{p}
Ssk1_[(D544)]-{p}
I
II
III
ON
OFF
C

Figure 2

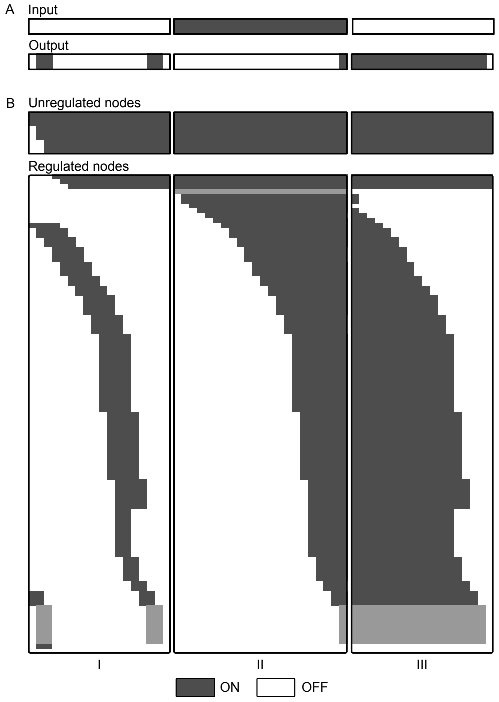

A
Input
Output
B
Unregulated nodes
Regulated nodes
I
II
III
ON
OFF

Figure 3

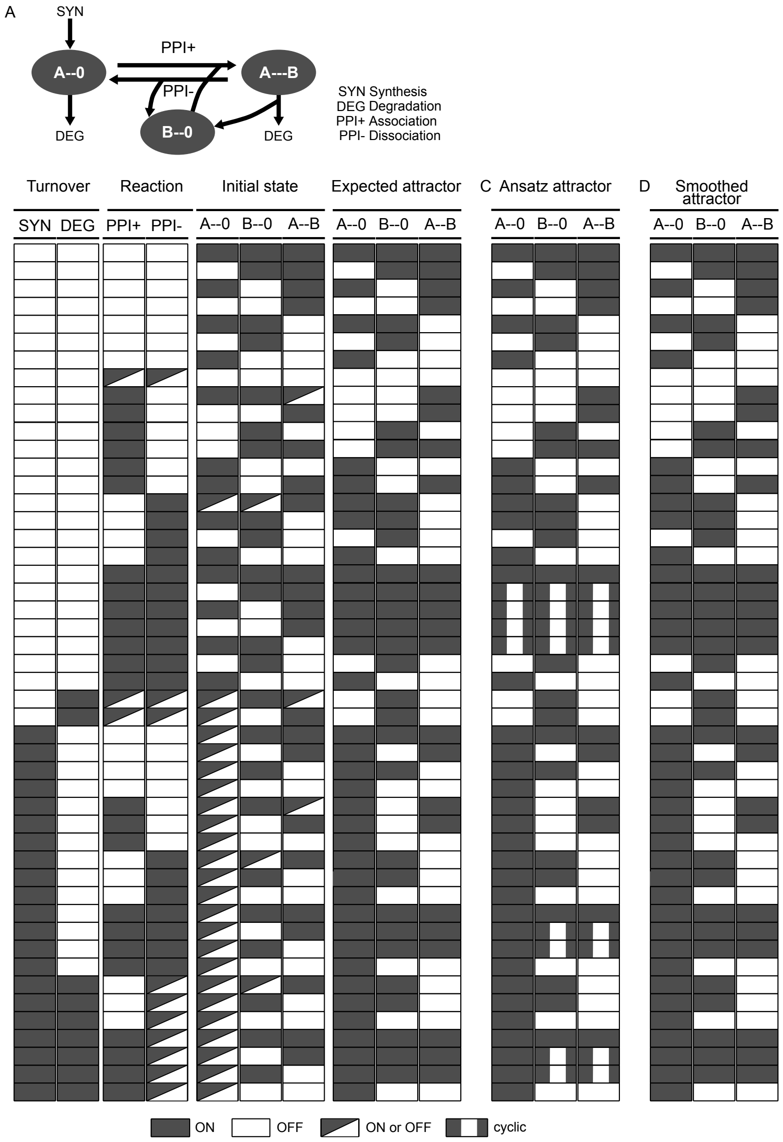

A
SYN
A--0
PPI+
A---B
PPI-
B--0
DEG
DEG
SYN Synthesis
DEG Degradation
PPI+ Association
PPI- Dissociation
Turnover
Reaction
Initial state
Expected attractor
C Ansatz attractor
D Smoothed attractor
SYN
DEG
PPI+
PPI-
A--0
B--0
A--B
A--0
B--0
A--B
A--0
B--0
A--B
A--0
B--0
A--B
ON
OFF
ON or OFF
cyclic

Figure S1

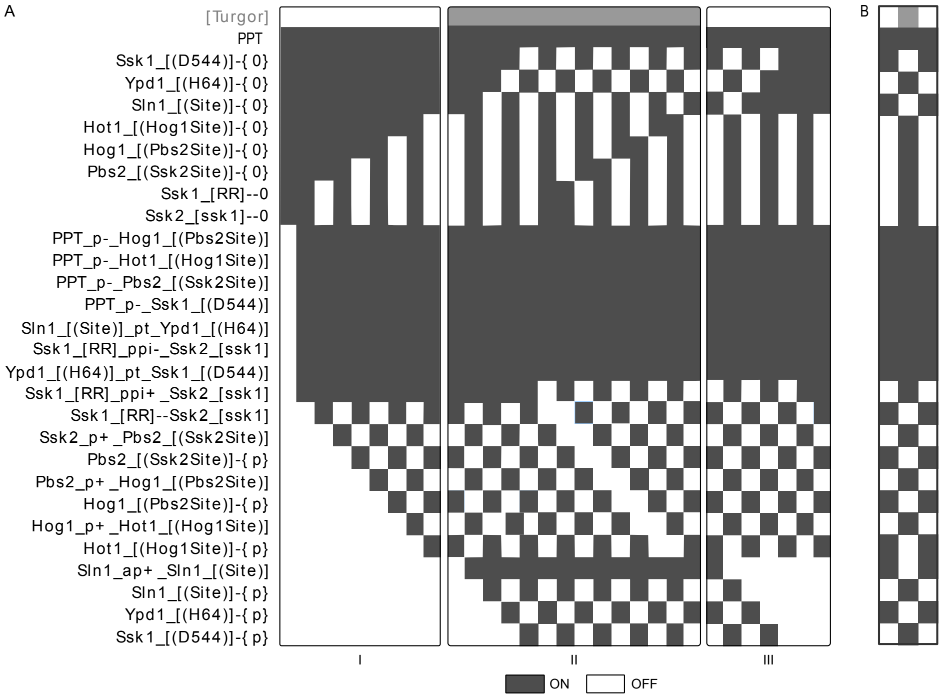

A
[Turgor]
PPT
Ssk1_[(D544)]-{0}
Ypd1_[(H64)]-{0}
Sln1_[(Site)]-{0}
Hot1_[(Hog1Site)]-{0}
Hog1_[(Pbs2Site)]-{0}
Pbs2_[(Ssk2Site)]-{0}
Ssk1_[RR]--0
Ssk2_[ssk1]--0
PPT_p-_Hog1_[(Pbs2Site)]
PPT_p-_Hot1_[(Hog1Site)]
PPT_p-_Pbs2_[(Ssk2Site)]
PPT_p-_Ssk1_[(D544)]
Sln1_[(Site)]_pt_Ypd1_[(H64)]
Ssk1_[RR]_ppi-_Ssk2_[ssk1]
Ypd1_[(H64)]_pt_Ssk1_[(D544)]
Ssk1_[RR]_ppi+_Ssk2_[ssk1]
Ssk1_[RR]--Ssk2_[ssk1]
Ssk2_p+_Pbs2_[(Ssk2Site)]
Pbs2_[(Ssk2Site)]-{p}
Pbs2_p+_Hog1_[(Pbs2Site)]
Hog1_[(Pbs2Site)]-{p}
Hog1_p+_Hot1_[(Hog1Site)]
Hot1_[(Hog1Site)]-{p}
Sln1_ap+_Sln1_[(Site)]
Sln1_[(Site)]-{p}
Ypd1_[(H64)]-{p}
Ssk1_[(D544)]-{p}
I
II
III
B
ON
OFF

Figure S2

Y