Bioinformatics, 33(7), 2017, 1040–1048
doi: 10.1093/bioinformatics/btw747
Advance Access Publication Date: 5 January 2017
Original Paper

Systems biology

Advanced Boolean modeling of biological networks applied to systems pharmacology

Itziar Irurzun-Arana*, José Martín Pastor, Iñaki F. Trocóniz*,† and José David Gómez-Mantilla†

Pharmacometrics & Systems Pharmacology, Department of Pharmacy and Pharmaceutical Technology, School of Pharmacy, University of Navarra, Pamplona 31008, Spain

*To whom correspondence should be addressed
†The authors wish it to be known that, in their opinion, the last two authors should be regarded as Joint Last Authors

Associate Editor: Alfonso Valencia

Received on March 17, 2016; revised on November 4, 2016; editorial decision on November 18, 2016; accepted on November 22, 2016

Abstract

Motivation: Literature on complex diseases is abundant but not always quantitative. Many molecular pathways are qualitatively well described but this information cannot be used in traditional quantitative mathematical models employed in drug development. Tools for analysis of discrete networks are useful to capture the available information in the literature but have not been efficiently integrated by the pharmaceutical industry. We propose an expansion of the usual analysis of discrete networks that facilitates the identification/validation of therapeutic targets.

Results: In this article, we propose a methodology to perform Boolean modeling of Systems Biology/Pharmacology networks by using SPIDDOR (Systems Pharmacology for effIcient Drug Development On R) R package. The resulting models can be used to analyze the dynamics of signaling networks associated to diseases to predict the pathogenesis mechanisms and identify potential therapeutic targets.

Availability and Implementation: The source code is available at https://github.com/SPIDDOR/SPIDDOR.

Contact: itzirurzun@alumni.unav.es, itroconiz@unav.es

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Computational models are frequently used in the area of biomedicine to interpret, describe or predict dynamic profiles associated to disease progression or drug effects. Among them, the so called population pharmacokinetic/pharmacodynamics (popPK/PD) models integrate different type of information, mainly, dosing paradigms and drug exposure, response data and patient characteristics to account for the time course of drug effects. PopPK/PD models are well established in clinical practice and drug development to individualize dosing, identify covariates responsible of inter-patient variability, and dose selection (Admiraal *et al.*, 2014; Borrat *et al.*, 2013; Buil-Bruna *et al.*, 2016). However, there are several pending challenges in the application of computational models to drug development such as early target identification, choice of best promising drug combinations, understanding resistance development and highlighting patient sub-population sensitive and non-sensitive to a particular therapeutic strategy.

To achieve these goals, popPK/PD models would require a greater mechanistic structure. Nonetheless, mechanistic models require large number of kinetic/dynamic parameters and the task of identifying these parameters is not always possible due to the lack of longitudinal and quantitative data available.

The emergent field of Systems Pharmacology (SP) has the role of bridging System Biology with popPKPD models and it is expected to help in overcoming the bottlenecks highlighted before (Bai *et al.*, 2014; Geerts *et al.*, 2015; Goryanin and Goryachev, 2011; Iyengar *et al.*, 2012; Lu *et al.*, 2014; Palmér *et al.*, 2014; van der Graaf and Benson, 2011; Wang *et al.*, 2015). SP models can be viewed as networks, which are simplified representations of biological systems in which the components of the system such as genes, proteins or metabolites are represented by nodes and the interactions between them by edges (Berger and Iyengar, 2009; Zhao and Iyengar, 2012). In general, two different approaches can be used to analyze this type of models: continuous dynamic methods, where the concentrations/amounts of the components are based on differential equations, or discrete dynamic strategies, in which each node can be characterized by only a few discrete states, indicated in contexts where quantitative and longitudinal data are scarce or even not available.

Boolean network models, originally introduced by Kauffman (Kauffman, 1993, 1969), represent the simplest discrete dynamic models. Very briefly, they only assume two discrete states for the nodes of a network, ON or OFF, corresponding to the logic values 1 (active) or 0 (not active, but not necessarily absent). That is why they are known as Boolean or logic models.

A well-designed logic model would be able to generate predictive outcomes given a set of initial conditions. In terms of applications, it would be possible to test how the elimination or overexpression of one or more components of the system affects the final state of the model, which may be useful in the design of combinatorial therapies for a disease or identification of essential components that could be tested as therapeutic targets. It could be also important to detect critical nodes whose perturbation leads to significant functional changes in the system in order to reduce the size of the network by removing the redundant components. This could be a starting point to try a more quantitative approach.

Currently the application of Boolean analysis to SP is still very limited, contrary to the case of applying dynamic models to continuous or non-continuous data, where there is a battery of tools to help the scientist for model implementation, fitting and evaluation (NONMEM, PsN, Pirana, etc.). Consequently, integration of discrete analysis tools in drug development has not been accomplished yet despite its great potential.

Based on these considerations we have developed a framework for an efficient Boolean analysis facilitating (i) model implementation and visualization, (ii) simulation of activation profiles associated with corresponding confidence intervals, (iii) attractor analysis and (IV) a system perturbation and sensitivity analysis. The tools presented in this manuscript consist on a set of comprehensive R scripts to perform discrete dynamic analysis in the context of development therapies for complex diseases.

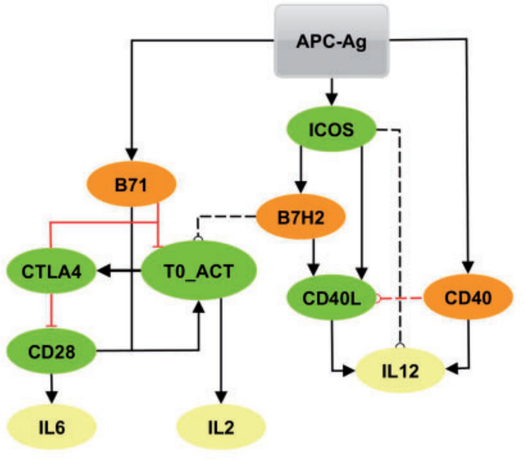


Fig. 1. Boolean network example with 12 nodes and 19 regulatory edges made with yEd Graph Editor software. Conceptual nodes (APC-Ag) are presented by a gray rectangle, whereas the molecules of the network are indicated with ellipses. Node colors reflect the nature of the molecules: APC molecules are shown in orange, T cell molecules are in green and interleukins appear in yellow (colored figure online). Arrowheads represent activation, red blunt edges indicate inhibition, black dashed lines imply positive modulations and red dashed lines are negative modulations. The description of the nodes and the Boolean functions is given in Table 1

From a methodological point of view the Boolean analysis presented in this work involves certain novelties. Common Boolean modeling approaches only define direct activation-inhibition relationships between the components of the network. In our models, new types of regulatory interactions have been introduced, the positive and negative modulations, which lead to richer dynamics between the nodes. We also propose a new option to perturb a component of the network emulating a 'polymorphism' of a node. Finally, novel approaches were developed for the exploratory analysis of the output of the simulations computed on these models: (i) we incorporate new visualization techniques to evaluate the attractors of the system and the effects of perturbations and (ii) a clustering method is used to group the nodes that lead to similar alterations within the network.

This article guides the reader through the tools developed in our laboratory for an example metabolic network (Fig. 1) based on a model for immune response to autoantigens (Ruiz-Cerdá *et al.*, 2016) and gives a feel of what can be done with its use. The package is called Systems Pharmacology for effIcient Drug Development On R (SPIDDOR). R scripts, help files and vignettes are available in https://github.com/SPIDDOR/SPIDDOR.

2 Methods

Our approach for Boolean modeling biological/pharmacological networks entails the workflow seen in Figure 2.

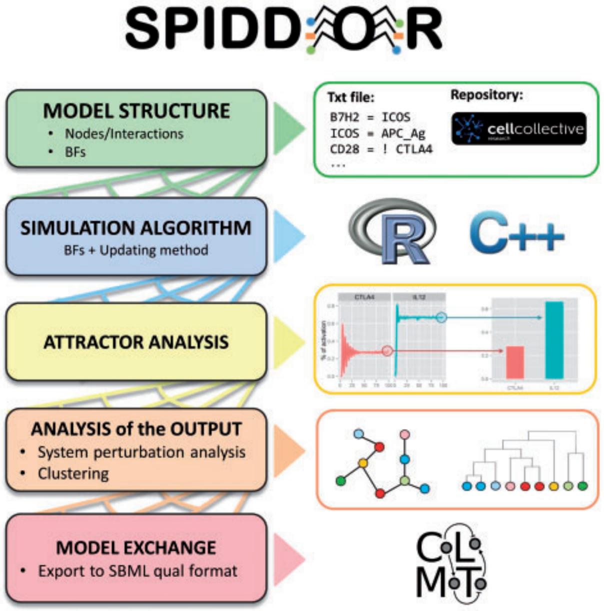


Fig. 2. Workflow of the methodology employed by SPIDDOR to perform Boolean modeling of biological networks and the most relevant outputs of each section. First, the model structure is defined in a text file or downloaded from The Cell Collective repository. Second, SPIDDOR reads the BFs from the input files and creates a simulation algorithm in R or C++. Then, the package is able to perform an attractor analysis and introduce perturbations into the model to analyze the output of these networks. Finally, the Boolean networks can be exported to SBML qual format to share models or use other platforms from the CoLoMoTo community

The first step in turning the concepts from literature into a discrete dynamic model is to represent the conceptual model as a directed graph showing the different nodes and the interactions between them. Such networks involve the coordinated interaction of many molecules and stimulus that include genes, proteins, metabolites, cellular states or other conceptual nodes as in Figure 1. After defining the components and interactions of the network, the next step is to implement the Boolean transfer functions based on an exhaustive literature research and introduce them in the R environment.

2.1 Boolean functions

The state of each node is determined by the state of its regulator nodes (nodes that control its activation/inhibition) based on transition rules known as the Boolean functions (BFs). Depending on the output of the BF, the state of a node can transit from one value to another as the simulation algorithm moves from an iteration to the next. Here, an iteration finishes when all the nodes in the network are updated according to their BFs (in many research works these iterations of the algorithm are referred as time steps but we prefer the term iteration to emphasize that a time step is not necessarily equivalent to a time length). BFs consist on a set of rules specifying how the nodes' states change over time, as a function of the current or past values of its regulator nodes. The main operators of Boolean dynamics are the conjunction AND, the disjunction OR and the negation NOT. Additionally, some convenience operators have been defined. For example, some nodes may need longer activation times of its regulator nodes to be activated. We represent this feature with the $\cap$ notation that can be seen in the BFs of Table 1 (Thakar *et al.*, 2007), and we called it threshold operator. The threshold operator requires a duration argument which indicates the number of previous iteration that must be evaluated for a regulator node. In the case of 1, it is used to represent that *CTLA-4* molecule is active only if the T cell activation node (*T0_ACT*) is ON for a defined number of iterations indicated by the parameter *T0_ACTmax* (*T0_ACTmax* = *3* in our simulations).

Generally, Boolean functions represent simple dynamics of activation and inhibition between nodes. In this work, we present two new possible combinations of Boolean operators that allow us to characterize more precisely some typical processes of biological systems. There are many cases in the literature in which a node A is not able to activate another node B, but A can increase or prolong B expression if B is activated by other signals. We considered this relationship as a positive modulation of node B by node A and we expressed it with the following combination of Boolean operators: *B** = *Activators OR (B AND A)*. As can be seen, this regulatory function introduces a self-regulation of the target node. Similarly, if node A cannot directly inhibit node B but it can decrease or shorten its expression it was considered as a negative modulation and we expressed it like *B*=Activators AND NOT (B AND A)*. Furthermore, we have designed these modulatory interactions between nodes to last only a few iterations: *B** = *Activators OR ((B AND A) AND NOT* ($\cap_{i=1}^{MOD} B^{t-1}$ *AND* $\cap_{i=1}^{MOD} A^{t-1}$)), with *MOD* argument specifying the maximum number of iterations that the positive modulation will last. All BFs corresponding to the example network are listed in Table 1.

BFs are introduced in SPIDDOR by a simple text file written with the appropriate equation semantics and the system transforms this file into R or C++ code. Another possibility is to load Boolean expressions from a pre-built network from The Cell Collective repository(Helikar *et al.*, 2012), a web-based platform included in the CoLoMoTo (Consortium for Logical Models and Tools) consortium (Naldi *et al.*, 2015).

2.2 Nodes updating

The outcome of a Boolean model is also influenced by the chosen updating method, which could be synchronous or asynchronous (Harvey and Bossomaier, 1997; Saadatpour *et al.*, 2010; Thakar *et al.*, 2007; Wang *et al.*, 2012). The updating method refers to the process of computing the BF of a node to activate or deactivate it in a particular iteration. In a synchronous updating method, the state of the network at each step is determined by the state of the nodes in the prior iteration of the algorithm. In such models, the dynamic trajectory of the network is deterministic, that is, the network will always reach the same state after the same number of iterations. This scheme assumes that all biological processes of the system have similar timescales, which seems quite unrealistic because molecular events are not coordinated in time. A more complex but realistic

Table 1. Boolean functions of the nodes of the network of Figure 1

Node	Description	Boolean Function	Text file
APC-Ag	Antigen presentation	APC-Ag* = APC-Ag	APC-Ag = APC-Ag
B71	CD80 molecule	B71* = APC-Ag	B71 = APC-Ag
ICOS	Inducible T-cell co-stimulator	ICOS* = APC-Ag	ICOS = APC-Ag
CD40	CD40 molecule	CD40* = APC-Ag	CD40 = APC-Ag
B7H2	ICOS ligand	B7H2* = ICOS	B7H2 = ICOS
CD28	CD28 molecule	CD28* = NOT CTLA4	CD28 =! CTLA4
CTLA4	Cytotoxic T-lymphocyte-associated protein 4	CTLA4*= $\cap_{i=1}^{T0\_ACTmax}$ T0_ACT$^{t-\mathrm{i}}$	CTLA4= THR_T0_ACT[3]
CD40L	CD40 ligand	CD40L* = ICOS AND B7H2 AND NOT (CD40 AND CD40L)	CD40L = ICOS & B7H2 &! (CD40 & CD40L)
T0_ACT	Activated T cell	T0_ACT* = (CD28 AND B71) OR (T0_ACT AND B7H2) AND NOT ($\cap_{i=1}^{MOD}$T0_ACT$^{t-\mathrm{i}}$ & $\cap_{i=1}^{MOD}B7H2^{t-i}$) AND NOT (CTLA4 AND B71)	T0_ACT = ((CD28 & B71) | (T0_ACT & B7H2) &! (MOD_T0_ACT & MOD_B7H2))) &! (CTLA4 & B71)
IL2	Interleukin 2	IL2* = T0_ACT	IL2 = T0_ACT
IL6	Interleukin 6	IL6* = CD28	IL6 = CD28
IL12	Interleukin 12	IL12* = (CD40 AND CD40L) OR (IL12 AND ICOS) AND NOT ($\cap_{i=1}^{MOD}$IL12$^{t-\mathrm{i}}$ & $\cap_{i=1}^{MOD}$ICOS$^{t-\mathrm{i}}$)	IL12 = (CD40 & CD40L) | (IL12 & ICOS) &! (MOD_IL12 & MOD_ICOS)

The *denotes the future state of a node.

strategy is the random asynchronous method, where the nodes of the system are updated according to the last update of their regulator nodes, which could be either in the previous or current iteration. In this method, the order in which the nodes update their states is selected randomly during each iteration. This introduces variability into the model, because the same initial conditions can lead to different final states of the network and with different time courses. Once the BFs are determined, they are implemented in the R environment. A function is written for each BF of the Boolean model, using both the synchronous and asynchronous updating methods. The R script containing the BFs for the example network of Figure 1 can be found in the Supplementary Material.

2.3 Network evolution in time

We developed a simulation algorithm to calculate the evolution of the network states taking into account the synchronous and asynchronous updating methods, although we recommend the use of the latter as it constitutes a more realistic approach as discussed above. The output of the algorithm is a matrix called *pattern.m* which represents the states of the nodes (ON/OFF) in each step (Fig. 3A). It must be stressed that, due to the stochasticity involved in the asynchronous updating scheme, the simulations must be computed repeated times in order to estimate an average of the dynamic trajectory of the network. This allows the calculation of the activation profiles of the nodes for any set of initial conditions. A schematic representation of this process is shown in Figure 3B.

To estimate 95% Confidence Intervals (CIs) for the activation profiles of the nodes, we used a method to calculate CIs for proportions by using a binomial distribution described by (Newcombe, 1998). For a more detailed description of this method see Supplementary Methods.

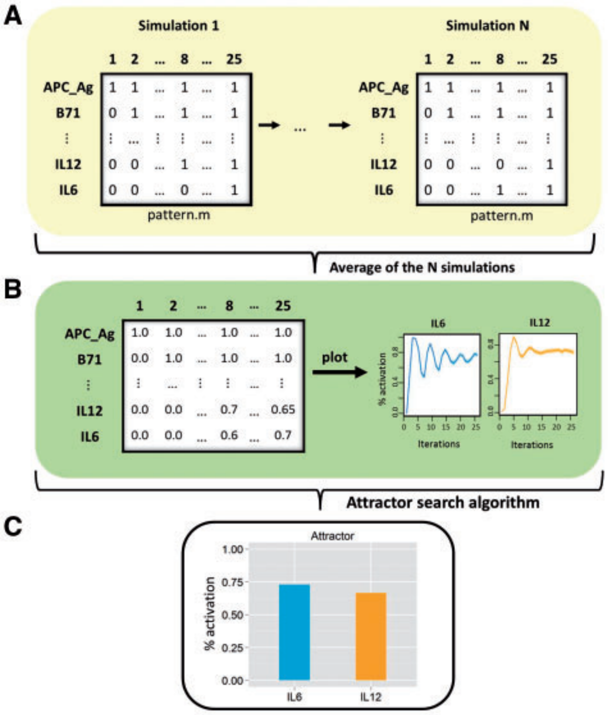


Fig. 3. Schematic representation of the steps performed by the asynchronous algorithm. The rows in the matrices correspond to the nodes of the network in Figure 1 and the columns to the iterations performed by the algorithm, 25 in this case. (**A**) The output of the simulation algorithm, the pattern.m matrix. (**B**) Average of the simulation algorithm results. The average was computed under 2000 (N) simulations in order to calculate the activation profiles of the nodes. (**C**) Probability of being ON of IL6 and IL12 nodes in the complex attractor found with the asynchronous attractor search algorithm

2.4 Attractor analysis

Starting from an initial condition, Boolean models eventually evolve into a limited set of stable states known as attractors (Hopfensitz *et al.*, 2012). Once the model has settled onto an attractor, it will remain there for the rest of the simulation. Attractors fall into three groups:

- Fixed-points, which consist of a single attractor state. They are the same for both synchronous and asynchronous update methods because of their time-independence property (Saadatpour *et al.*, 2010).
- Simple or limit cycles are set of states in which the system regularly oscillates. These are typical of the synchronous method where each state has only one possible successor state. In our models, however, the states of the nodes in the current iteration not only depend on the states of the nodes in the previous step, but also on prior steps due to the temporal predicates implemented with the threshold operator and the modulators introduced in the system. This produces regular cycles with duplicated states that we called ‘complex cycles’.
- Complex attractors are set of states in which the system irregularly oscillates due to the randomness involved in asynchronous networks. In these models there is usually more than one possible successor state for each state, so the system does not oscillate in cycles.

Generally, large-scale or highly interconnected networks converge into a complex attractor when an asynchronous updating scheme is used. This oscillatory behavior in Boolean models is due to the presence of negative feedback loops in the network (Saadatpour and Albert, 2013; Thomas and D’ari, 1990; Thomas *et al.*, 1995). Attractors in moderate size networks are often linked to cellular steady states, cell cycles, circadian rhythms or to phenotypes (Akman *et al.*, 2012; Bilsland *et al.*, 2014; Li *et al.*, 2004; Sun *et al.*, 2014). However, it is difficult to make biological inferences from complex attractors as they normally include a high number of stable states that do not oscillate in single cycles.

Our algorithm to identify attractors with the synchronous updating method starts from an initial state and repeatedly performs state transitions until an already visited state is reached. When the synchronous attractors are found (a fixed-point, a simple cycle or a complex cycle) they can be visualized as transition tables where the color inside the table represents the ON/OFF states of the nodes (Supplementary Fig. S1). Asynchronous attractor search is more complex as it computes the attractor via exhaustive repetitions of the simulation algorithm. The states in asynchronous attractors do not oscillate cyclically, so they cannot be visualized using transition tables as in the previous case. For this reason, we decided to summarize the information about all the stable states in the attractor by generating the probability that a given node is ON inside the complex attractor. Finally, we visualize these probabilities using bar graphs (Fig. 3C).

Identification of all the attractors in large-scale asynchronous models is an arduous task due to the computational time required, especially if the attractors are complex because some of their states rarely occur. Moreover, these steady states can change when initial conditions are modified or perturbations are included in the system. We found that the activation probabilities of the nodes in complex attractors almost did not change if the 'unusual' states were ignored, suggesting that we could estimate an approximation of the attractor by excluding those rare states from the analysis. This approximation decreases the number of repetitions needed for the asynchronous attractor search algorithm. In addition, for large-scale networks, we recommend coding the simulation algorithm on C++ to increase speed up to 60-fold. We used the Rcpp R package to communicate R to the C++ algorithm and get the result back to the R environment, allowing its use by normal R users not skilled in C++.

Our main attractor search algorithm is coded to identify the attractor for a given initial condition. We introduced this simplification because we were not interested in testing all the possible initial states, as we typically defined a few possible initial conditions for our networks. However, in some cases there is not enough information to specify the initial condition of a system and sampling of a multitude of initial conditions is necessary. For those interested in this feature, SPIDDOR includes an attractor search algorithm that searches the attractors for networks with less than 20 nodes, as the number of initial conditions to test grow exponentially with the number of nodes. For larger networks, we allow the specification of a subset of nodes (always less than 20) in which all the combinations are to be tested, or the specification of a number of starting states to test (the restriction of maximum 20 nodes limits the initial conditions to test in less than 100 000).

Since the hypothetical network used in this article is moderated in size, there is no need of using a parallelized algorithm to reduce the computing time for attractor searching. Even so, this feature is contemplated in our framework and the code for the parallelization using the snowfall library (Knaus, 2009) is included in the github repository.

2.5 Perturbation analysis

A system perturbation analysis can be performed in order to evaluate which node knockouts or overexpressions lead to significant variations of the network dynamics. A knockout implies the deactivation of a component during all the simulation, whereas an overexpression generates a persistent activation of a node. Another possibility is to overexpress a node but only after its first activation or to activate/deactivate a node for some time. This analysis allows the researcher to model the effects of pharmacological blockades or simulate targeted therapies such as monoclonal antibodies (mAbs).

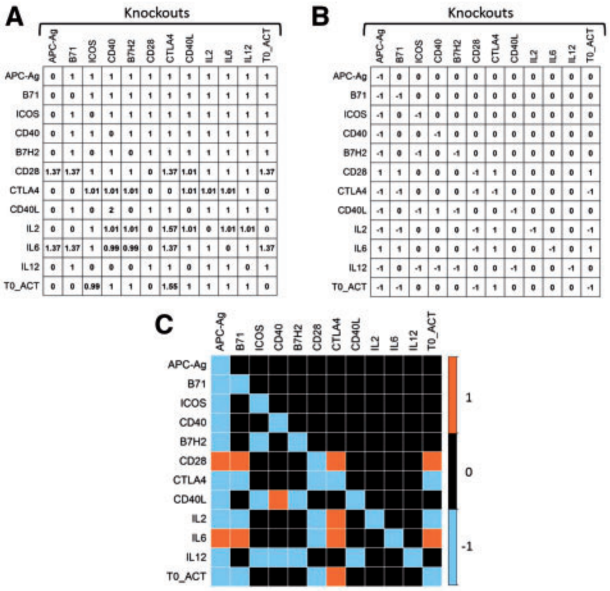
Our modeling approach also allows the emulation of 'polymorphism like' alterations on the components of the network that can result in modifications of their activation patterns. In biology, genetic polymorphisms cause decreased, increased, or absent gene expression or molecular activity by multiple mechanisms. We included these 'mutation like' perturbations in which the activity of a node is associated with a probability dependent on the 'polymorphism like' conditions. In other words, when a polymorphism was included, we decreased the activity of a node to a lower extent (75%, 50%, 25%...). In this way, when a polymorphism of 50% activity was introduced in a node, this node was activated only 50% of the times in which its regulator nodes were activated.

The activation level of the nodes in normal conditions and when a node was knocked-out or overexpressed were compared in order to analyze how the perturbation of single nodes affected the stable patterns of the rest of the nodes in the network. If the probability of being ON for a node was decreased due to the inclusion of a perturbation, it means that the perturbation caused a lower activation of the component compared to the unperturbed condition. Conversely, if the probability was increased due to a perturbation, it indicates that the perturbation caused a higher activation of the component.

We developed a perturbation analysis algorithm that performs combined synchronous–asynchronous simulations for faster identification of attractors with or without perturbations. First, the program initiates a synchronous attractor search in order to detect whether the network reaches a fixed-point, as this type of attractor is the same in both synchronous and asynchronous algorithms. If this is not the case, we run the asynchronous attractor search to find the complex attractor and the frequency of being ON of each node in these attractors that represent its activation level.

The result of the perturbation analysis is a square matrix in which the number of rows and columns is equal to the number of nodes in the network. It indicates how the knockout/overexpression of the 'column node' affected each 'row node' (Fig. 4A). The value in each cell of the matrix corresponds to the probability ratio between the perturbed and the normal conditions. We call to this ratio the *Perturbation Index*(PI) of the nodes. The equation for a given node i under a perturbation in J is the following: $PI\_J_i = Prob(i)_{Perturbation_J}/Prob(i)_{Normal}$, where $Prob$ is the probability of being ON of the node in a given attractor state. Values close to 1 mean that the activity of a node in normal and altered conditions was very similar, and therefore the perturbation had a minor effect on the component.

In order to improve the visualization of this analysis, we transformed the resulting matrix to store only 3 possible values, -1, 0 and 1, as shown in Figure 4B. The -1 substitutes the positions where there is a lower activation of a component (value < 0.8), the 0 indicates no significant variation between the perturbed and unperturbed conditions, and the 1 represents the locations where there is a higher activation of a node (value > 1.25). If a more complex network is being modeled, it is preferable to use more than 3 values to take into account different levels of regulations. The rescaled matrix can be represented using the corrplot package in R in order to



A

	Knockouts											
	APC-Ag	B71	ICOS	CD40	B7H2	CD28	CTLA4	CD40L	IL2	IL6	IL12	T0_ACT
APC-Ag	0	1	1	1	1	1	1	1	1	1	1	1
B71	0	0	1	1	1	1	1	1	1	1	1	1
ICOS	0	1	0	1	1	1	1	1	1	1	1	1
CD40	0	1	1	0	1	1	1	1	1	1	1	1
B7H2	0	1	0	1	0	1	1	1	1	1	1	1
CD28	1.37	1.37	1	1	1	0	1.37	1.01	1	1	1	1.37
CTLA4	0	0	1.01	1.01	1.01	0	0	1.01	1.01	1.01	1	0
CD40L	0	1	0	2	0	1	1	0	1	1	1	1
IL2	0	0	1	1.01	1.01	0	1.57	1.01	0	1.01	1.01	0
IL6	1.37	1.37	1	0.99	0.99	0	1.37	1	1	0	1	1.37
IL12	0	1	0	0	0	1	1	0	1	1	0	1
T0_ACT	0	0	0.99	1	1	0	1.55	1	1	1	1	0

B

	Knockouts											
	APC-Ag	B71	ICOS	CD40	B7H2	CD28	CTLA4	CD40L	IL2	IL6	IL12	T0_ACT
APC-Ag	-1	0	0	0	0	0	0	0	0	0	0	0
B71	-1	-1	0	0	0	0	0	0	0	0	0	0
ICOS	-1	0	-1	0	0	0	0	0	0	0	0	0
CD40	-1	0	0	-1	0	0	0	0	0	0	0	0
B7H2	-1	0	-1	0	-1	0	0	0	0	0	0	0
CD28	1	1	0	0	0	-1	1	0	0	0	0	1
CTLA4	-1	-1	0	0	0	-1	-1	0	0	0	0	-1
CD40L	-1	0	-1	1	-1	0	0	-1	0	0	0	0
IL2	-1	-1	0	0	0	-1	1	0	-1	0	0	-1
IL6	1	1	0	0	0	-1	1	0	0	-1	0	1
IL12	-1	0	-1	-1	-1	0	0	-1	0	0	-1	0
T0_ACT	-1	-1	0	0	0	-1	1	0	0	0	0	-1

Fig. 4. Results arising from a knockout analysis of the network in Figure 1 and the subsequent steps to improve its visualization. (**A**) Numeric matrix with the corresponding *Perturbation indexes* in each cell; (**B**) Ranking of the values from matrix A; (**C**) Heatmap of matrix B in which the color indicates if the node knockout entails a lower (blue) or higher (orange) activation of a component compared to an unperturbed simulation (colored figure online)

visualize the individual values contained in the matrix as colors (Fig. 4C). In this work, we only performed single node disruptions, altering one-by-one each node from the network, but double or triple perturbations can also be induced in the simulations.

2.6 Clustering

Hierarchical clustering methods (Hartigan, 1975) determine clusters of similar data points based on their distance and build a hierarchical structure on top of them. We applied this method on the results of the perturbation analysis, under the assumption that node alterations that provoke similar effects on the rest of the nodes of the system will cluster together. Here, we employed the Euclidean metric to determine the distances between each node Perturbation Index and, as merging approach, we used the average-linkage strategy. For example, the distance between a knockout in node A and a knockout in node B would be calculated as follows:

$$d(A,B)=\sqrt{\sum_{i=1}^{n}(PI\_B_i-PI\_A_i)^2}$$

where n is the number of nodes in the network and PI_A and PI_B are the *Perturbation indexes* of the nodes under the knockout in A and B respectively. The results of this exercise are summarized as heatmaps complemented by dendrograms that illustrate the similarity between the perturbations of the system (see Results and Discussion).

2.7 Model interoperability

Over the years, different software supporting logical models have been developed, generating different formats to store these models. To address this problem, a novel model exchange format, called SBML qual (Systems Biology Markup Language Qualitative Models) (Chaouiya *et al.*, 2013), was developed by the CoLoMoTo community (Naldi *et al.*, 2015). SBML qual is designed for the representation of multivalued qualitative models of biological networks, thus, enabling models to be shared and used with other platforms and tools without the need of rewriting them in a different format.

We developed a function to export the networks evaluated with SPIDDOR to SBML qual format. We note that SBML does not support networks with temporal operators, so the converter removes these patterns from the nomenclature to store them adequately. In such cases, the output of the simulations performed with SPIDDOR will differ from the results obtained with other platforms because the temporal operators notably change the dynamic evolution of the network.

Models encoded in SBML qual can be submitted to the BioModels database (Le Novere *et al.*, 2006) and to The Cell Collective and GINsim (Gonzalez *et al.*, 2006) software repositories.

3 Results and discussion

In the current work we present the SPIDDOR package which is specifically tailored to the design and analysis of Boolean network models in the area of SP. There are already several software tools and packages available for Boolean modeling of biological systems like BooleanNet (Albert *et al.*, 2008), BoolNet (Müssel *et al.*, 2010), SimBoolNet Zheng *et al.* (2010), ChemChains Helikar and Rogers (2009), or GINsim (Gonzalez *et al.*, 2006). We note that SPIDDOR differs from other existing methodologies in the following characteristics:

a. Positive and negative modulations: Apart from the basic activation-inhibition interactions, two new regulatory connections have been introduced in the Boolean models, the positive and negative modulations. In the BFs of Table 1, IL12 cytokine is positively modulated by the ICOS molecule, meaning that ICOS only activates IL12 if IL12 has already been activated by another regulator node. ICOS does not work as a complete activator because it cannot activate IL12 by itself but it can intensify another activating signal, therefore working as a 'sustainer'. Similarly, the concept of negative modulator is applied to the CD40 node which does not prevent the activation of node CD40-L by itself but can lessen its expression. Figure 5 shows how the activation probabilities of nodes IL12 and CD40L changed when their corresponding logic functions were modified. Both graphs changed when modulation interactions were included compared to simple activations or inhibitions, reflecting the importance of choosing the proper BF for a component. The advantage of incorporating these relationships is that they provide a more semi-quantitative representation of the activity between components, allowing the inclusion of more biologically realistic interactions.

b. Polymorphisms: SP models could be employed to test multiple scenarios as for example the different disease evolution or response to treatment among subject with diverse polymorphisms in a single or various nodes. This perturbation varies a node activity from 0 to 1 and checks the effect of these variations on a desired outcome. In Figure 6 it is shown how polymorphisms acting on the activity of B71 node decreased the activation levels of T0_ACT and increased the expression of IL6 compared to normal response. This type of analysis can be used to evaluate gene mutations that are linked to a particular disease, and test which polymorphisms could trigger similar molecular alterations as the ones reported for the disease. This perturbation analysis is complementary to the introduction of node knockouts or overexpressions which are not realistic representations of human physiopathology. Furthermore, the introduction of a node knockout could provoke a total blockage of one or several pathways hindering the analysis of less severe perturbation or complementary perturbations in other nodes. A similar analysis could be used to explore the effect of target engagement on drug treatment. For example, in Supplementary Figure S2, when a

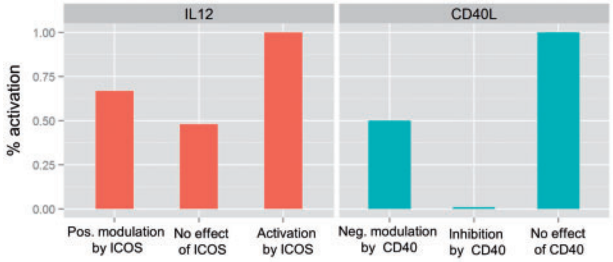


Fig. 5. Activation probability of IL12 (left) and CD40L (right) nodes with different Boolean functions. The probability of being ON for IL12 varies when ICOS makes a positive modulation (BF of Table 1), a complete activation (IL12 = (CD40 & CD40L) | ICOS) or has no effect on IL12 (IL12 = CD40 & CD40L). On the other hand, the output of CD40L changes when we introduce a negative modulation by CD40 node (CD40L= ICOS & B7H2 &! (CD40 & CD40L)), a complete inhibition by CD40 (CD40L= ICOS & B7H2 &! CD40L) or when CD40 has no effect on CD40L (CD40L= ICOS & B7H2)

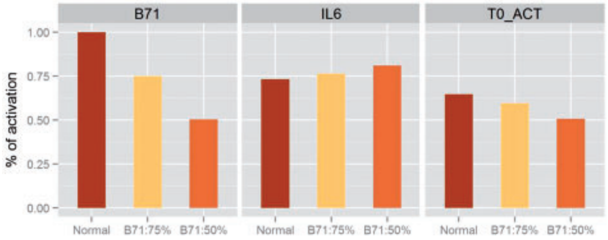


Fig. 6. Relative expression profiles of B71, IL6 and T0_ACT with different levels of polymorphisms acting on B71. A polymorphism was simulated on B71 node to reduce a 25% and a 50% its activity. This perturbation increased the levels of IL6 expression and decreased the activation of T cell T0_ACT node

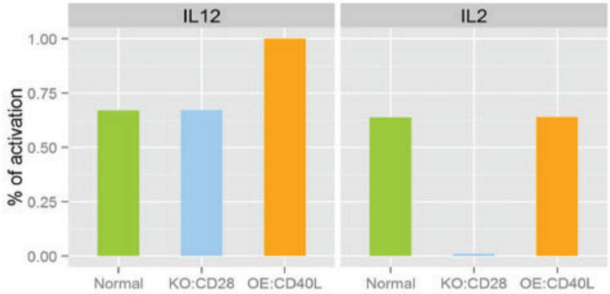


Fig. 7. Activation levels of IL2 and IL12 with different perturbations of the system. Two different perturbations were introduced in the model, knock-out of node CD28 (KO:CD28) and over-expression of node CD40L (OE:CD40L), in order to see how the probability of being ON of IL2 and IL12 change

mAb is used as a therapeutic agent, it can be studied what is the required level of target inhibition for an anti-Icos mAb to achieve a reduction of 50% on IL12 expression. Similarly, if a polymorphism is introduced on an input node by setting it to a specific average level of activation, it is possible to explore different background noise levels on the system and evaluate the impact of environmental fluctuations (Domedel-Puig *et al.*, 2011).

c. Visualization of attractor states: Some of the mentioned discrete modeling tools like Boolnet have functions to visualize the complex attractors as interconnected graphs representing state transitions inside the attractor. However, when the number of nodes in the network is high, these graphs are extremely difficult to analyze and may not provide meaningful information for the scientists who are not familiar with such discrete outputs. For this reason, we improved their visualization by representing the activation probability of the nodes using bar graphs. For the attractor analysis of the example network we simulated the evolution of the system under a continuous antigen presentation (APC-Ag = 1) in synchronous and asynchronous mode. Under the synchronous updating method, we found a 'complex cycle' composed of 28 states (Supplementary Fig. S1). The asynchronous attractor search algorithm with 1000 simulation steps and repeated 16 times found a complex attractor composed of 84 states whose activation probabilities are summarized in Supplementary Table S1. In Figure 7, we simulated network perturbations by introducing a knockout on CD28 molecule and an overexpression of CD40L and analyze how these alterations affect to IL2 and IL12 activation probabilities.

d. Visualization of perturbation analysis: Several tools were developed for the exploratory analysis of the network output to evaluate many nodes perturbations at the same time on the attractors of the system and to cluster them according to the effects that they provoke. In Figure 8A it is shown how a knockout on APC-Ag node modifies the activation probability of all the nodes in the network (shown in orange and blue) as it is the input node of the system, while a knockout on ICOS molecule only downregulates B7H2, CD40L and IL12 components (shown in blue). This is quite easy to infer by observing the structure of the network in Figure 1, but in larger systems the effects of the manipulations are not so easily deduced. The result of the hierarchical clustering calculation is displayed as a dendrogram in the top of these heatmaps (Fig. 8). For example, the effects of ICOS and B7H2 knockouts in the system are very similar, so they are clustered together in the dendrogram of Figure 8A.

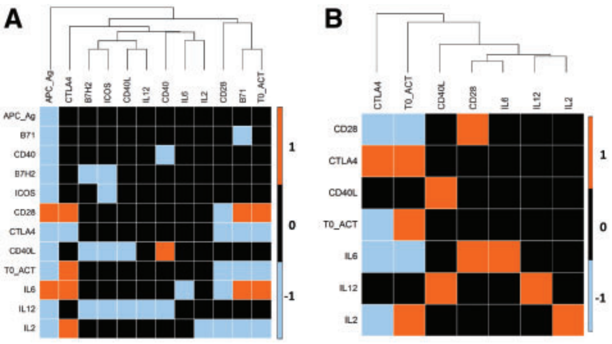


Fig. 8. Hierarchical clustering of the perturbations induced on the nodes of the network in Figure 1. Heatmaps indicate the effect of single perturbations (knockouts on the left and overexpressions on the right) on the nodes of the network. The perturbations that lead to a higher activation of the nodes compared to an unperturbed situation are represented in orange while a lower activation of the nodes is indicated in blue

The dynamic perturbation analysis is a technique used to identify critical nodes and facilitate network validation. In this type of modeling frameworks, it is possible to emulate a disease on the biological network under study by changing the initial conditions of the computer simulations. Thanks to the dynamic perturbation method it is possible to test which perturbations can revert the disease condition (Saadatpour *et al.*, 2011). Such results could be used to prioritize which of the knockouts or constitutive activations should be studied first in wet bench experiments.

Another practical use of the visualization technique explained above is the possibility of performing a sensitivity analysis of the network to discover which nodes have a higher impact on other components of the system. In the matrices of Figure 4, nodes ICOS, CD40, CD40L and B7H2 have a higher influence on IL12 interleukin, as their perturbation lead to a significant downregulation of the molecule. If the interest lies mainly in the response of this component, a network reduction could be applied by removing the components that do not regulate IL12. This ability is important to reduce the size of complex SP networks. This sensitivity analysis can be complemented by the use of the polymorphism tool introduced before to identify sources of interindividual variability by highlighting the nodes which polymorphisms are more likely to provoke large changes in specific outputs.

The immune network presented in this work is an illustrative example used to describe the new methodologies which application we consider useful in the SP field. The results obtained from our simulations should not be considered as a full representation of the immune response because many immunological components have been left apart for simplification purposes.

Despite the advantages of Boolean networks and the methodologies presented in this work, some limitations need to be considered. First, it is important to realize that manually building a biological network may be time-consuming and (inevitably) subjective as BFs are established following the researcher criteria. Some tools exist to infer networks automatically from experimental data (Scutari, 2010) but different algorithms lead to different networks while different networks are generally deduced from different datasets, therefore, it is also subjective which algorithm and dataset to use.

A main limitation lies in the reliability of these models. In this project, we tried to reproduce the experimental observations discussed on the research articles used to build the network. The heatmaps created with our framework are useful for this task. For example, in the heatmap of Figure 8A, we can see how a knockout in CD28 leads to a lower activation of IL2, which is consistent with the results found in the literature (Howland *et al.*, 2000). However, this could be a complex task when there is a lack of information about the nodes under study. We are currently working on new possible validation methods based on microarray or RNA-seq analysis, but further work needs to be done.

4 Conclusion

Computational models have been increasingly used to support drug development and are widely accepted by scientific community and even for regulatory purposes. A key challenge when using these powerful approaches is to match the right model with the right questions in a particular research context. Although Boolean networks cannot be used for precise estimations such as drug dosing in pediatric or renal impairment population, they are useful to gain insight into the qualitative behavior of a system under study. This is especially relevant for large scale systems in which a detailed kinetic characterization of the system is not feasible due to data restrictions or limited knowledge. More precise quantitative models require exponentially more complex and quality data, and sometimes, acquisition of such data could be restricted by technical constraints, as is the case of immunology in which there are not available techniques for the continuous *in-vivo* measurement of cells subpopulations and cytokines in different tissues. Therefore, it is mandatory to get the best use of the available knowledge in each stage of development, for which it is essential to explore the full potential of tools like Boolean networks. We consider that the methodologies presented in this work can potentiate the use of Boolean networks in SP by introducing versatile tools to enrich the analysis of these systems.

Funding

II and JDG were financed by Janssen Research and Development.

Conflict of Interest: none declared.

References

Admiraal,R. *et al.* (2014) Towards evidence-based dosing regimens in children on the basis of population pharmacokinetic pharmacodynamic modelling. *Arch. Dis. Child.*, **99**, 267–272.

Akman,O.E. *et al.* (2012) Digital clocks: simple Boolean models can quantitatively describe circadian systems. *J. R. Soc. Interface*, **9**, 2365–2382.

Albert,I. *et al.* (2008) Boolean network simulations for life scientists. *Source Code Biol. Med.*, **3**, 16.

Bai,J.P.F. *et al.* (2014) Systems pharmacology modeling: an approach to improving drug safety. *Biopharm. Drug Dispos.*, **35**, 1–14.

Berger,S.I. and Iyengar,R. (2009) Network analyses in systems pharmacology. *Bioinformatics*, **25**, 2466–2472.

Bilsland,A.E. *et al.* (2014) Mathematical model of a telomerase transcriptional regulatory network developed by cell-based screening: analysis of inhibitor effects and telomerase expression mechanisms. *PLoS Comput. Biol.*, **10**, e1003448.

Borrat,X. *et al.* (2013) Modeling the influence of the A118G polymorphism in the OPRM1 gene and of noxious stimulation on the synergistic relation between propofol and remifentanil: sedation and analgesia in endoscopic procedures. *Anesthesiology*, **118**, 1395–1407.

Buil-Bruna,N. *et al.* (2016) Establishing the quantitative relationship between lanreotide autogel®, chromogranin a, and Progression-Free survival in patients with nonfunctioning gastroenteropancreatic neuroendocrine tumors. *AAPS J*, **18**, 703–712.

Chaouiya,C. *et al.* (2013) SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.*, **7**, 135.

Domedel-Puig,N. *et al.* (2011) Information routing driven by background chatter in a signaling network. *PLoS Comput. Biol.*, **7**, e1002297.

Geerts,H. *et al.* (2015) Assessing the synergy between cholinomimetics and memantine as augmentation therapy in cognitive impairment in schizophrenia. A virtual human patient trial using quantitative systems pharmacology. *Front. Pharmacol.*, **6**, 198.

Gonzalez,A.G. *et al.* (2006) GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems*, **84**, 91–100.

Goryanin,I.I. and Goryachev,A.B. (2011) *Advances in Systems Biology*. Springer Science & Business Media, Springer-Verlag New York.

Hartigan,J.A. (1975) *Clustering Algorithms*, 99th edn. John Wiley & Sons, Inc., New York, NY, USA.

Harvey,I. and Bossomaier,T. (1997). Time out of joint: Attractors in asynchronous random boolean networks. In: *Proceedings of the Fourth European Conference on Artificial Life*, pages 67–75.

Helikar,T. and Rogers,J.A. (2009) ChemChains: a platform for simulation and analysis of biochemical networks aimed to laboratory scientists. *BMC Syst. Biol.*, **3**, 58.

Helikar,T. *et al.* (2012) The cell collective: toward an open and collaborative approach to systems biology. *BMC Syst. Biol.*, **6**, 96.

Hopfensitz,M. *et al.* (2012) Attractors in Boolean networks: a tutorial. *Comput. Stat.*, **28**, 19–36.

Howland,K.C. *et al.* (2000) The roles of CD28 and CD40 ligand in T cell activation and tolerance. *J. Immunol.*, **164**, 4465–4470.

Iyengar,R. *et al.* (2012) Merging systems biology with pharmacodynamics. *Sci. Transl. Med.*, **4**, 126ps7.

Kauffman,S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.

Kauffman,S.A. (1993) *The Origins of Order: Self Organization and Selection in Evolution*. Oxford University Press, USA.

Knaus, J. *et al.* (2009) Easier parallel computing in R with snowfall and sfCluster. *The R Journal*, **1**, 54–59.

Le Novere,N. *et al.* (2006) BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, **34**, D689–D691.

Li,F. *et al.* (2004) The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. U. S. A.*, **101**, 4781–4786.

Lu,Y. *et al.* (2014) Use of systems pharmacology modeling to elucidate the operating characteristics of SGLT1 and SGLT2 in renal glucose reabsorption in humans. *Front. Pharmacol.*, **5**, 274.

Müssel,C. *et al.* (2010) BoolNet–an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, **26**, 1378–1380.

Naldi,A. *et al.* (2015) Cooperative development of logical modelling standards and tools with CoLoMoTo. *Bioinformatics*, **31**, 1154–1159.

Newcombe,R.G. (1998) Two-sided confidence intervals for the single proportion: comparison of seven methods. *Stat. Med.*, **17**, 857–872.

Palmér,R. *et al.* (2014) Effects of il-1*β*–blocking therapies in type 2 diabetes mellitus: a quantitative systems pharmacology modeling approach to explore underlying mechanisms. *CPT Pharmacometrics Syst. Pharmacol.*, **3**, e118.

Ruiz-Cerdá,M.L. *et al.* (2016) Towards patient stratification and treatment in the autoimmune disease lupus erythematosus using a systems pharmacology approach. *Eur. J. Pharm. Sci*, **94**, 46–58.

Saadatpour,A. and Albert,R. (2013) Boolean modeling of biological regulatory networks: a methodology tutorial. *Methods*, **62**, 3–12.

Saadatpour,A. *et al.* (2010) Attractor analysis of asynchronous Boolean models of signal transduction networks. *J. Theor. Biol.*, **266**, 641–656.

Saadatpour,A. *et al.* (2011) Dynamical and structural analysis of a T cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia. *PLoS Comput. Biol.*, **7**, e1002267.

Scutari,M. (2010) Learning Bayesian networks with the bnlearn R package. *J. Stat. Softw.*, **35**, 1–22.

Sun,Z. *et al.* (2014) Multi-level modeling of light-induced stomatal opening offers new insights into its regulation by drought. *PLoS Comput. Biol.*, **10**, e1003930.

Thakar,J. *et al.* (2007) Modeling systems-level regulation of host immune responses. *PLoS Comput. Biol.*, **3**, e109.

Thomas,R. and D'ari,R. (1990) *Biological Feedback*. CRC press, Boca Raton, Florida.

Thomas,R. *et al.* (1995) Dynamical behaviour of biological regulatory networks—i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull. Math. Biol.*, **57**, 247–276.

van der Graaf,P.H. and Benson,N. (2011) Systems pharmacology: bridging systems biology and pharmacokinetics-pharmacodynamics (PKPD) in drug discovery and development. *Pharm. Res.*, **28**, 1460–1464.

Wang,R.S. *et al.* (2012) Boolean modeling in systems biology: an overview of methodology and applications. *Phys. Biol.*, **9**, 055001.

Wang,Y. *et al.* (2015) Multi-scale modeling of cell survival and death mediated by the p53 network: a systems pharmacology framework. *Mol. Biosyst*,11, 3011–3021.

Zhao,S. and Iyengar,R. (2012) Systems pharmacology: network analysis to identify multiscale mechanisms of drug action. *Annu. Rev. Pharmacol. Toxicol.*, **52**, 505–521.

Zheng,J. *et al.* (2010) SimBoolNet—a cytoscape plugin for dynamic simulation of signaling networks. *Bioinformatics*, **26**, 141–142.