BMC
Systems Biology

METHODOLOGY ARTICLE   **Open Access**

SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools

Claudine Chaouiya[1*], Duncan Bérenguier[2], Sarah M Keating[3†], Aurélien Naldi[4], Martijn P van Iersel[3],
Nicolas Rodriguez[3,5], Andreas Dräger[6,7], Finja Büchel[7], Thomas Cokelaer[3], Bryan Kowal[16], Benjamin Wicks[8],
Emanuel Gonçalves[3], Julien Dorier[9], Michel Page[10,11], Pedro T Monteiro[1,12], Axel von Kamp[13], Ioannis Xenarios[9],
Hidde de Jong[10], Michael Hucka[14], Steffen Klamt[13], Denis Thieffry[15], Nicolas Le Novère[3,5],
Julio Saez-Rodriguez[3*] and Tomáš Helikar[16*]

Abstract

Background: Qualitative frameworks, especially those based on the logical discrete formalism, are increasingly used to model regulatory and signalling networks. A major advantage of these frameworks is that they do not require precise quantitative data, and that they are well-suited for studies of large networks. While numerous groups have developed specific computational tools that provide original methods to analyse qualitative models, a standard format to exchange qualitative models has been missing.

Results: We present the Systems Biology Markup Language (SBML) Qualitative Models Package ("qual"), an extension of the SBML Level 3 standard designed for computer representation of qualitative models of biological networks. We demonstrate the interoperability of models via SBML qual through the analysis of a specific signalling network by three independent software tools. Furthermore, the collective effort to define the SBML qual format paved the way for the development of LogicalModel, an open-source model library, which will facilitate the adoption of the format as well as the collaborative development of algorithms to analyse qualitative models.

Conclusions: SBML qual allows the exchange of qualitative models among a number of complementary software tools. SBML qual has the potential to promote collaborative work on the development of novel computational approaches, as well as on the specification and the analysis of comprehensive qualitative models of regulatory and signalling networks.

* Correspondence: chaouiya@igc.gulbenkian.pt; saezrodriguez@ebi.ac.uk;
thelikar2@unl.edu
†Equal contributors
[1]Instituto Gulbenkian de Ciência, Rua da Quinta Grande 6, 2780-156 Oeiras,
Portugal
[3]European Molecular Biology Laboratory, European Bioinformatics Institute
(EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD,
UK
[16]Department of Biochemistry, University of Nebraska-Lincoln, Lincoln, NE
68588, USA
Full list of author information is available at the end of the article

BioMed Central

Background

Studies by S. Kauffman [1] and R. Thomas [2] founded the logical discrete approach to model biological molecular networks and analyse their behaviours. In these networks, components (e.g., genes or proteins) assume discrete values representing their activity levels (e.g., gene expression). Components are connected by directed edges that embody regulatory (causal) effects, forming an influence network. The activity level of each component evolves depending on the activity levels of the components influencing it. The rules that determine component activity levels are defined in terms of logical rules or functions, corresponding to the underlying biological/biochemical regulatory mechanisms. The dynamical behaviour of the network is then generated by evolving the component levels following a specific updating scheme (e.g., synchronous, asynchronous or stochastic). The dynamics can subsequently be represented in terms of a state transition graph, where the nodes represent (discrete) states of the model, while the edges denote transitions between these states.

While other mathematical frameworks, including differential equations, can be used to model biological processes in great detail, the logical formalism is particularly suitable for the modelling of large networks for which precise kinetic data are not available. In fact, logical models have become increasingly popular. They have been recently used to model complex dynamical behaviours and provide insights into numerous biological systems, including gene regulatory networks (e.g., [3-6]), signal transduction (e.g., [7-14]), as well as cell cycle (e.g., [15-18]), in species ranging from bacteria and viruses (e.g., [3,19,20]) to yeast (e.g., [17,21-23]), flies (e.g., [24-26]), plants (e.g., [27,28]), and to humans (e.g., [11-13,18,29]).

Often based on qualitative knowledge of regulatory mechanisms and published data, discrete models can be assembled through a "bottom-up" approach, whereby each logical function represents specific, biological interactions between the components of the network. Recently, "top-down" approaches have been applied to construct logical models by automatic inference from high-throughput experiments (e.g., [30]).

Many simulation and analysis software tools for logical models already exist, including ADAM [31], BoolNet [32], BooleanNet [33], Cell Collective [34,35] CellNetAnalyzer [36], CellNOpt [30], ChemChains [37], GINsim [38], Odefy [39], SimBoolNet [40], and SQUAD [41].

The state transition graphs describe the discrete dynamics of networks and thus embody their dynamical properties. However, these graphs may quickly become too large and difficult to analyse. This has led several groups to propose the use of model-checking techniques [42] that explore, for instance, attractors (e.g., stable states or terminal cycles) and paths leading to them [43]. A number of logical modelling tools allow properties of the state transition graphs to be verified by means of existing model-checking tools, such as NuSMV [44-47]. The properties are formulated in terms of temporal logic or in a suitable high-level query template capturing recurrent biological questions [48]. The model checker tests if the state transition graph, which may be explicitly generated or implicitly encoded in a symbolic description of the model, satisfies the property. For example, while GINsim exports symbolically encoded logical models into SMV files, BIOCHAM integrates NuSMV [47] providing an interface for the specification and verification of properties expressed in several temporal logics [46]. A detailed description of the use of model-checking techniques in the context of qualitative models of biological networks is outside the scope of this paper, but see reference [49] for a review and additional examples.

Over the years, different formats have been developed to store logical models, ranging from simple text files containing truth tables and/or logical functions to XML-based file formats. Standards such as the Systems Biology Markup Language (SBML [50]) or the Systems Biology Graphical Notation (SBGN [51]) have been developed to enable unified exchange of biological/biochemical molecular maps. SBML supports process-based mathematical frameworks with a reaction-centred description of biochemical processes. Because the building blocks of qualitative models are fundamentally different from species and reactions used in (core) SBML models, previous attempts to represent logical models in SBML led to a distorted use of the standard. Indeed, variables in Boolean networks, logical models and some Petri nets represent discrete levels of activities rather than amounts/numbers of molecules. Moreover, simulation of logical models do not generally imply the notion of a continuous time. Consequently, the processes involving them cannot be described as reactions *per se*, but rather as transitions between states.

SBML Level 3 is modular and thereby enables the development and inclusion of packages extending the core with additional features. Using this modular structure, we developed a novel Qualitative Models ("qual") package to support the standard definition and exchange of qualitative (discrete) models.

It is worth noting that, although SBML qual is currently mainly used for logical models, it was developed to support standard Petri nets as well, due to commonalities between the frameworks. While Petri nets are mostly used to study metabolic networks, they have also been employed to model regulatory and signalling networks (see reviews [52-54]). Until now, the Petri net community relied on specialised exchange formats (e.g., PNML, http://www.pnml.org) and simulation tools that support SBML core (e.g., [55]).

Given the open source nature of SBML qual and the collaborative nature of the SBML community, the new standard should be swiftly adopted and implemented in many existing tools supporting logical models and their relatives such as Petri nets. The cooperation on SBML qual further fostered synergistic efforts to articulate and improve existing tools, leading to the launching of the Common Logical Modelling Tools (CoLoMoTo) project (http://co.mbine.org/colomoto/), which gathers many groups developing and using logical modelling software tools.

Methods

Development of the qual package

A draft proposal of a SBML package to encode qualitative models was initially proposed in 2008. Between 2008 and 2012, the proposal was refined, through community consultations and dedicated meetings by developers of various related software tools, and in particular members of the CoLoMoTo project. In 2011, the proposal was accepted through a community vote. The final specification was accepted by the SBML Editors in the spring of 2013.

LibSBML & JSBML

LibSBML is an application programming interface (API) library for reading, writing, manipulating and validating content expressed in the SBML format [56]. It is written in ISO C and C++, provides language bindings for .NET, Java, Python, Perl, Ruby, MATLAB and Octave, and includes many features that facilitate the adoption and use of both SBML and the libSBML. JSBML, a pure Java library for SBML, provides an API that maps all SBML elements to a flexible and extended Java type hierarchy whilst striving for 100% compatibility with the libSBML Java API [57]. As of November 2013, libSBML supports SBML qual in its public release (from version 5.9) while JSBML supports the package in its development branch and will include support in its next major release. LibSBML and JSBML are freely available as source code and binaries for all major operating systems under the LGPL open source terms (see http://sbml.org/Downloads). JSBML has been integrated in the LogicalModel library (see Results).

Computer simulations

To demonstrate the interoperability of models via SBML qual, we analysed a specific signalling network using three different software tools briefly described below.

CellNOpt is an open-source software used for creating logic-based models of signal transduction networks [30]. CellNOpt consists of a set of R packages available in Bioconductor, which are also available via a Python wrapper, as well as a Cytoscape plug-in (CytoCopteR) which contains a SBML qual importer and exporter.

CellNOpt converts a network (a signed, directed graph) into a scaffold of all possible models compatible with the network and subsequently trains this scaffold with data [58]. It includes a variety of formalisms: (i) Boolean models, simulated via synchronous update or by computation of steady-states, (ii) semi-quantitative constrained Fuzzy logic, and (iii) ordinary differential equations (ODEs) derived from the logical model [30]. While the choice of a specific formalism depends on the data at hand, scope, and question, the followed workflow is similar. The network can be simplified by compressing nodes that are intermediates between perturbed or measured nodes. Links impinging on nodes that are not observable (with no readout downstream) or not controllable (with no perturbation upstream of them) are also taken aside as their status cannot be derived from the data.

CellNOpt generates logical models as hyper-graphs by adding all combinations of OR and AND gates that are compatible with the network (i.e., Sums of Products [59]). This leads to a hyper-graph representing a superposition of all Boolean models compatible with the initial network. Subsequently, an optimisation procedure is applied to find the combination of gates and the parameters that best explain the data, by minimising an objective function that quantifies the difference between data and simulation, while penalising model size. This provides an optimum model or, more generally, a family of optimal models. Optimisation can be performed using a built-in genetic algorithm, or using external optimisation packages; in particular CellNOpt is connected to Meigo [60]. Furthermore, CellNOpt can leverage Answer Set Programming to efficiently find all possible Boolean models via the software package caspo [61].

Once an optimal model (or family of models) has been generated, it can be analysed in various ways. For example, it can be simulated to predict the outcome of new experiments [58]. One can also analyse the properties of a family of models, or compare models obtained for different cell types [62]. One can also identify missing links in the network using the module CNORFeeder [63]. The flexibility of the scripting languages (R, or Python) simplifies the writing of analysis workflows, and Cytocopter enables combined analysis with other Cytoscape tools and plug-ins.

GINsim is a free (Java) software application devoted to the logical (multi-valued) modelling of regulatory and signalling networks [38,64]. It provides a user-friendly graphical interface to define models from scratch. Models can also be imported from different formats. GINsim supports the simulation of logical models and generates the resulting state transition graphs, considering a range of update policies (see below). GINsim also offers a number of functionalities to explore the dynamical properties of logical

models, some of which (e.g., determination of stable states) can be efficiently analysed without generating the complete network dynamics.

It is well known that regulatory circuits (or feedback loops) can generate crucial dynamical properties [65]: positive circuits (encompassing an even number of inhibitions) produce multi-stability whereas negative circuits (encompassing an odd number of inhibitions) underlie stable oscillations. To help analyse these properties, GINsim identifies all the regulatory circuits embedded into a network and compute the regions of the state space, called functionality contexts, where they generate the related property (multi-stability *versus* oscillations).

One can use various updating schemes to generate the dynamics of a logical model. When in a given state, several components are called to change their values, these updates can be done synchronously, asynchronously, or considering a priority scheme [15]. Under the synchronous scheme, all components are updated simultaneously, leading to one transition at most for each state and thus resulting in deterministic (linear) state transition sequences. Under the asynchronous scheme, single component updates are considered separately, assuming that underlying delays are different but unknown; consequently, alternative trajectories are often generated, giving rise to non-deterministic state transition graphs.

Of particular interest is the asymptotical dynamical behaviour of these models, which is captured by the notion of *attractors*. From a logical point of view, attractors take two forms: stable states, and terminal cyclic strongly-connected components (as defined in graph theory). Note that stable states and terminal elementary cycles (where in each state, a unique component is updated) are shared between synchronous and asynchronous updating schemes but this is not the case for other cyclic attractors. GINsim supports both the synchronous and asynchronous updating schemes, which can lead to rather distinct dynamical properties. In particular, asynchronous dynamics can be quite complex.

In this respect, Hierarchical Transition Graphs (HTG) provide a compact and informative view of the dynamics in the form of a graph where nodes embody sets of states that are either irreversible (denoting irreversible sequences of states) or strongly connected (denoting oscillations in the form of transient or terminal complex components). For more details on HTG, see reference [66].

Finally, to handle the analysis of large models, several groups have devised reduction methods [10,58,67,68]. In this respect, the last (beta) version of GINsim allows users to get rid of (pseudo-) output species that do not regulate other nodes or regulate only pseudo-output nodes. This reduction has no impact on the number, nature and reachability of the attractors and it is particularly efficient for signalling networks as shown with our example model (see Results section).

The Cell Collective is a web-based platform for the construction, simulation, and analysis of Boolean-based models [34,35]. The platform includes a Knowledge Base for users to annotate the models and keep track of experimental research papers associated with each interaction included in the model. Within the platform, models can be shared directly on the web or via download using the SBML qual format (as well as in the form of text files including the list of logical expressions, and as .csv files with truth tables).

Models constructed in Cell Collective are Boolean (each species has a Boolean function associated with it, and assumes either an active or inactive state), and simulations can also include stochastic elements. Furthermore, data input/output from the analyses are continuous, providing a semi-quantitative measure to better match modelling results with laboratory experiments [14,37]. At the input level, this is accomplished by assigning a probability of being active in time t to each external species (i.e., those with no regulators), in contrast to classical Boolean simulations where each external species is fixed to 0 or 1.

The activity level, or the probable active state of an output species, is measured by calculating the ratio of 0's and 1's over the last n time steps (n is configurable to any discrete value, [35,37]); this ratio (multiplied by 100) provides the activity level on the y-axis (e.g., see Results; these parameters can be changed through the user interface). In the case of real-time simulations, as a single simulation evolves in time, the activity level of each species in the model is calculated as the ratio of 0's and 1's within a predefined sliding window [35,37].

One of the assets of the Cell Collective is its user interface, which has been carefully designed to enable the construction of computational models in a non-technical fashion, in order to render modelling also amenable to non-modellers. That is, the construction of the models is based purely on qualitative knowledge about a particular regulatory mechanism (e.g., kinase X phosphorylates and activates species Y), without the need to manually enter Boolean expressions (these are created in the background based on the biological data provided [69]). Although creating relatively small Boolean models can be easily done by writing Boolean functions, defining models with species regulated by many regulators, or through complex regulatory mechanisms can often result in complex, nested functions (e.g., [14,29]), which can be cumbersome to define manually even for seasoned modellers.

Results and discussion

In this section, we present the SBML qual package and its validation by exchanging and interpreting a moderately complex signalling network model among three independent software tools. In addition, we illustrate the interest of model exchange by applying complementary

simulation and analysis features. The section ends with a description of the LogicalModel library.
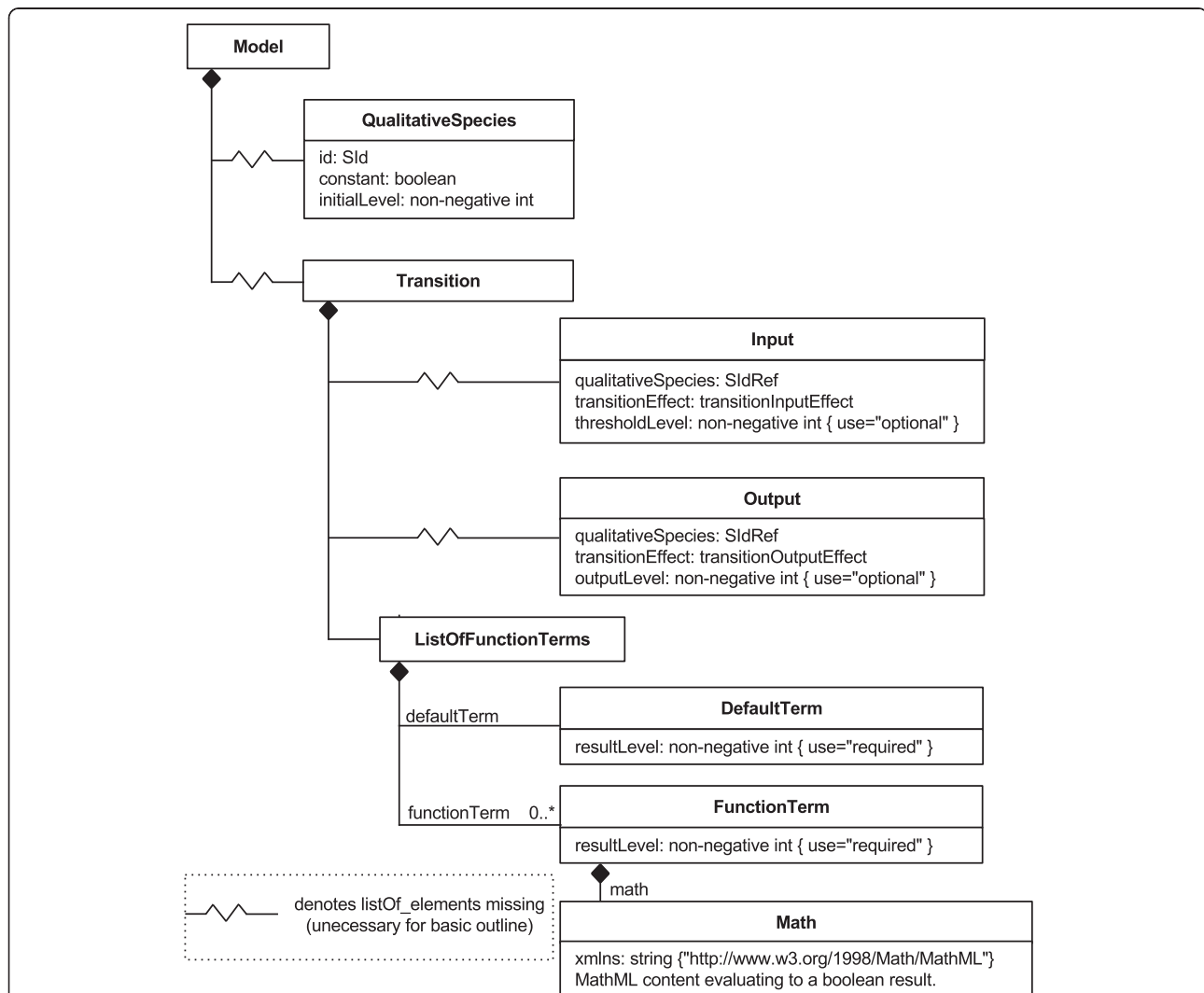
The SBML qual package

The SBML qual package extends the core SBML Level 3 standard, and enables standard exchange and interoperability of discrete models: logical models (Boolean and multilevel) and standard Petri nets. The full specification is available at http://identifiers.org/combine.specifications/sbml.level-3.version-1.qual.version-1.release-1 [70]. The structure of SBML qual is depicted in Figure 1. The rationale of the format relies on the features of qualitative models, with a (discrete) state space and event-driven state transition processes.

The main elements of a SBML qual document are *QualitativeSpecies,* representing the entities of the model as the molecular components of the network, and *Transitions* that contain the rules defining the state of given species at each iteration step. Here, the term transition is reminiscent of the Petri net terminology where a transition is enabled according to the state of its input places, and its firing modifies the state of its input and output places. For logical models, a transition defines the logical rule associated with a network component.

Each *QualitativeSpecies* assumes a discrete value (e.g., 0 or 1 for the Boolean case), and its definition bears an attribute *initialLevel* that specifies the value(s) at the beginning of the simulation, and an attribute *maxLevel* that specifies the maximal level allowed. For instance, *maxLevel* would be 1 in a Boolean model. As for the *Species* of SBML Core, a *QualitativeSpecies* is associated with a compartment.

Figure 1 Simplified UML diagram for the SBML L3 Qualitative Models package (SBML qual). The *QualitativeSpecies* represent the entities involved in the model. These are referenced as either *Inputs* or *Outputs* of the *Transition* element. A *Transition* describes the way the level of each *QualitativeSpecies* may be altered depending on the levels of other entities in the model (a complete UML diagram can be found in [70]).

A *Transition* comprises *Inputs* (*QualitativeSpecies* mentioned in the logical function), *Outputs* (species whose values at iteration $t + 1$ are determined by the logical rules evaluated at t) and *FunctionTerms*, containing conditions, as well as the values that the *Output* species will assume at $t + 1$, whenever a given condition is met. At each iteration, all *FunctionTerms* within a *Transition* are evaluated. The term evaluating to *true* dictates the resulting state and the *Output* species are updated accordingly at $t + 1$.

Each member of the *ListOfFunctionTerms* associated with a *Transition* contains a mathematical expression that returns a Boolean, as well as a *resultLevel* that indicates the level to be applied to the *Outputs* when this expression evaluates to true. A *defaultTerm* is also defined to establish the result when none of the *FunctionTerms* apply. The combined set of *defaultTerm* together with the list of *FunctionTerms* establish the state transitions for the entities involved. Figure 2 provides an illustration of a simple Boolean model encoded in SBML qual.

Demonstration of model interoperability

As part of the SBML extension development and approval process (by the SBML Editors) is the requirement that at least two independent software tools fully implement the proposed package. CellNOpt, GINsim, and the Cell Collective have been recently registered with the SBML community as tools currently supporting the *SBML qual* package. These tools have been further used to demonstrate how a logical model can be handled with different software tools using SBML qual as an exchange format.

As the three aforementioned software tools can provide different perspectives on the dynamics of discrete/ logical models, this section is organised to demonstrate their complementarity. More specifically, we present a conceptual pipeline that enables scientists to derive a discrete model from high-throughput data, conduct thorough analyses, and ultimately use the model to further guide experiments. CellNOpt is used to derive a logical model via a top-down approach, exploiting
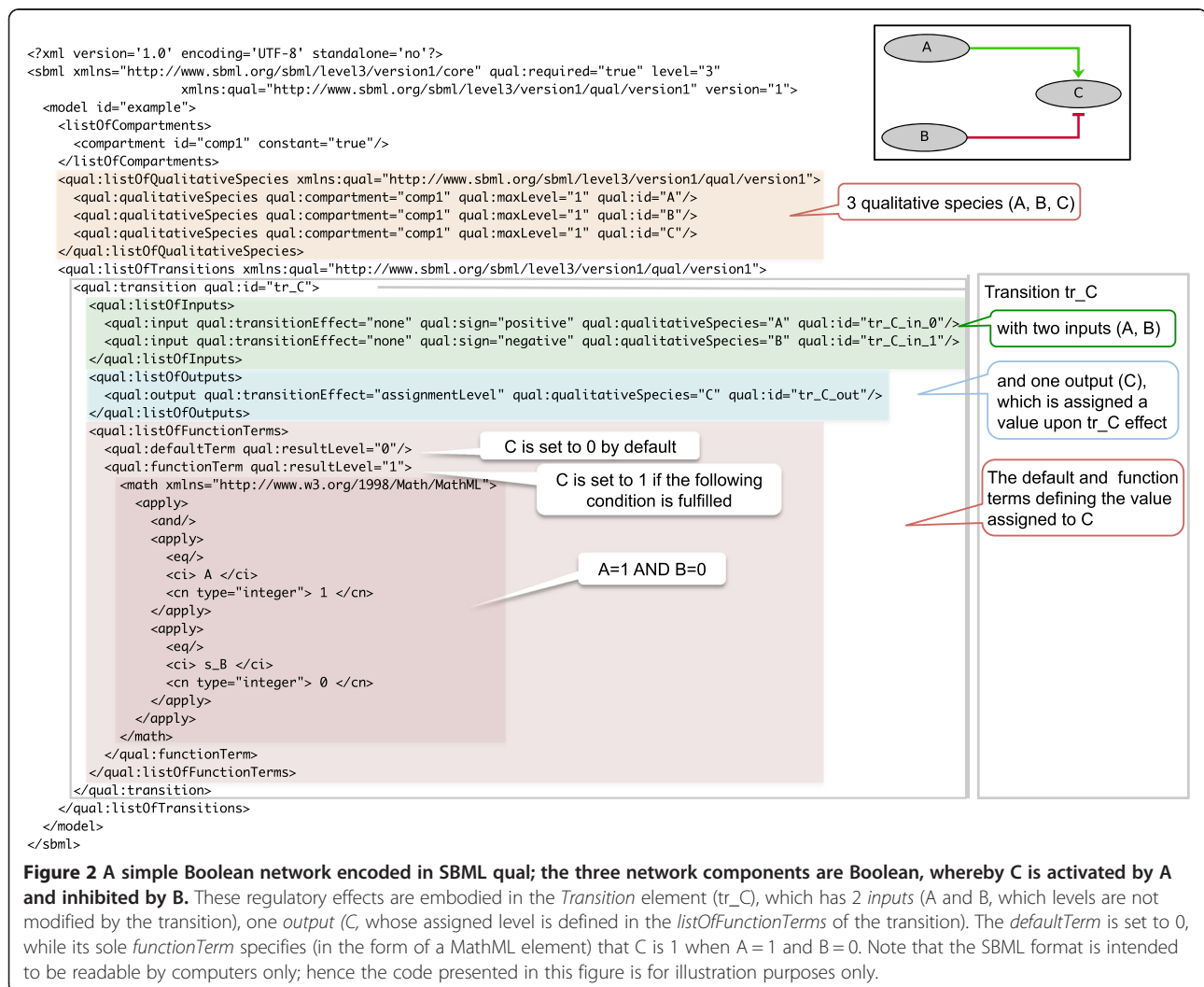

Figure 2 A simple Boolean network encoded in SBML qual; the three network components are Boolean, whereby C is activated by A and inhibited by B. These regulatory effects are embodied in the *Transition* element (tr_C), which has 2 *inputs* (A and B, which levels are not modified by the transition), one *output* (C, whose assigned level is defined in the *listOfFunctionTerms* of the transition). The *defaultTerm* is set to 0, while its sole *functionTerm* specifies (in the form of a MathML element) that C is 1 when A = 1 and B = 0. Note that the SBML format is intended to be readable by computers only; hence the code presented in this figure is for illustration purposes only.

experimental high-throughput data and an initial qualitative description of a signalling network (see Methods). Inferred models are subsequently simulated and analysed using additional techniques implemented in GINsim and the Cell Collective.

Generation of a EGF/TNFα discrete model with CellNOpt

The focus of CellNOpt is to utilise experimental data to generate logical models based on prior knowledge on signalling pathways (Prior Knowledge Networks, PKNs). The example model used herein is based on a PKN that combines two important mammalian signalling pathways induced by the Epidermal Growth Factor (EGF) and Tumour Necrosis Factor alpha (TNFα). EGF and TNFα ligands stimulate ERK, JNK and p38 MAPK cascades, the PI3K/AKT pathways, and the NFκB cascade. In addition, the network encompasses cross-talks between these pathways, as well as two negative feedback loops: one in the NFκB cascade and one in the MAPK cascade. Note that this network was previously used in [71] to illustrate a variety of logical modelling approaches using synthetic data. Here, however, we slightly modified this PKN by adding an autocatalytic feedback loop on the phosphatase (ph) regulating the activation of SOS-1 (Figure 3).

The PKN was subsequently trained using the synchronous update Boolean simulation, in combination with the CNORFeeder package to obtain the optimal logical model (see Methods) used as an example in this paper (Figure 3). Instead of using experimental data, an ordinary differentiation equation (ODE) model representing the "true network" was employed to generate the data and train the PKN. These data (in the form of time series) were thus obtained by simulating the ODE model upon stimulation of EGF and TNFα, and inhibition of PI3K and Raf-1 in different combinations. The readout nodes (i.e., the proteins whose activities were measured upon stimulation) are highlighted in Figure 3. To reflect imprecisions in our knowledge of biological pathways, the topology of the data generator model ("golden standard") is slightly different from the PKN. More precisely, a link from Map3K7 to MKK7 has been omitted in the PKN, to which an extra edge from PI3K to Map3K1 was further added. A workflow with CellNOpt was able to recover the "golden standard" model from this PKN and the experimental data (see http://www.cellnopt.org/doc/cnodocs/examples_sbml.html for more detail). This final model was then exported to SBML qual and simulated and analysed using all three tools.
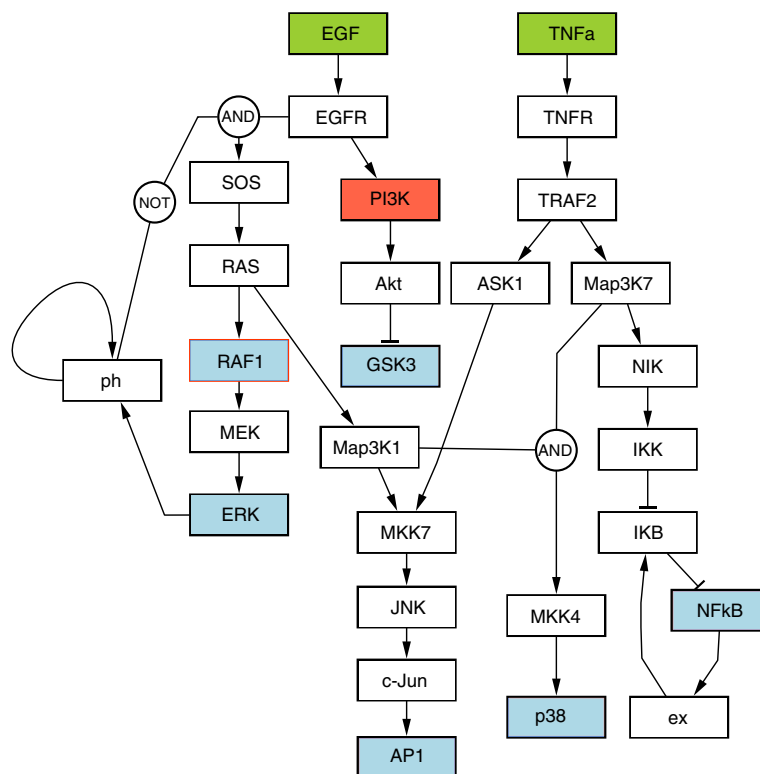
Figure 3 Boolean model obtained by CellNOpt and visualised using the Activity Flow language of the Systems Biology Graphical Notation [51] and drawn with CySBGN [72]. Different colours define the experimental design of the data used to train the model: (i) green boxes denote external stimuli, (ii) red boxes correspond to species blocked by kinase inhibitors, and (iii) blue boxes denote species that were measured (readouts).

Dynamical cross-validation of the model

Dynamical (synchronous) simulations of the EGF/TNFα network for the four different initial conditions gave consistent results with the three software tools. In this respect, Figure 4 shows the consistent global state evolution, attractor reachability, as well as temporal evolution of selected nodes for two of these conditions (see Additional file 1 for a full set of simulation results). Depending on the initial condition, simulations result in one of two stable states or in one of two cycles encompassing six states.

Model analyses in GINsim

As mentioned in the previous section, starting from the null state, simulations using synchronous updating of the example model result in a unique attractor for each of the four combinations of the two external inputs (EGF and TNFα), either a stable state or a simple terminal cycle (Additional file 1). Although we expect to get the same attractors under the asynchronous update, their reachability may be affected. Moreover, the number of states possibly visited before reaching an attractor may greatly differ between synchronous and asynchronous

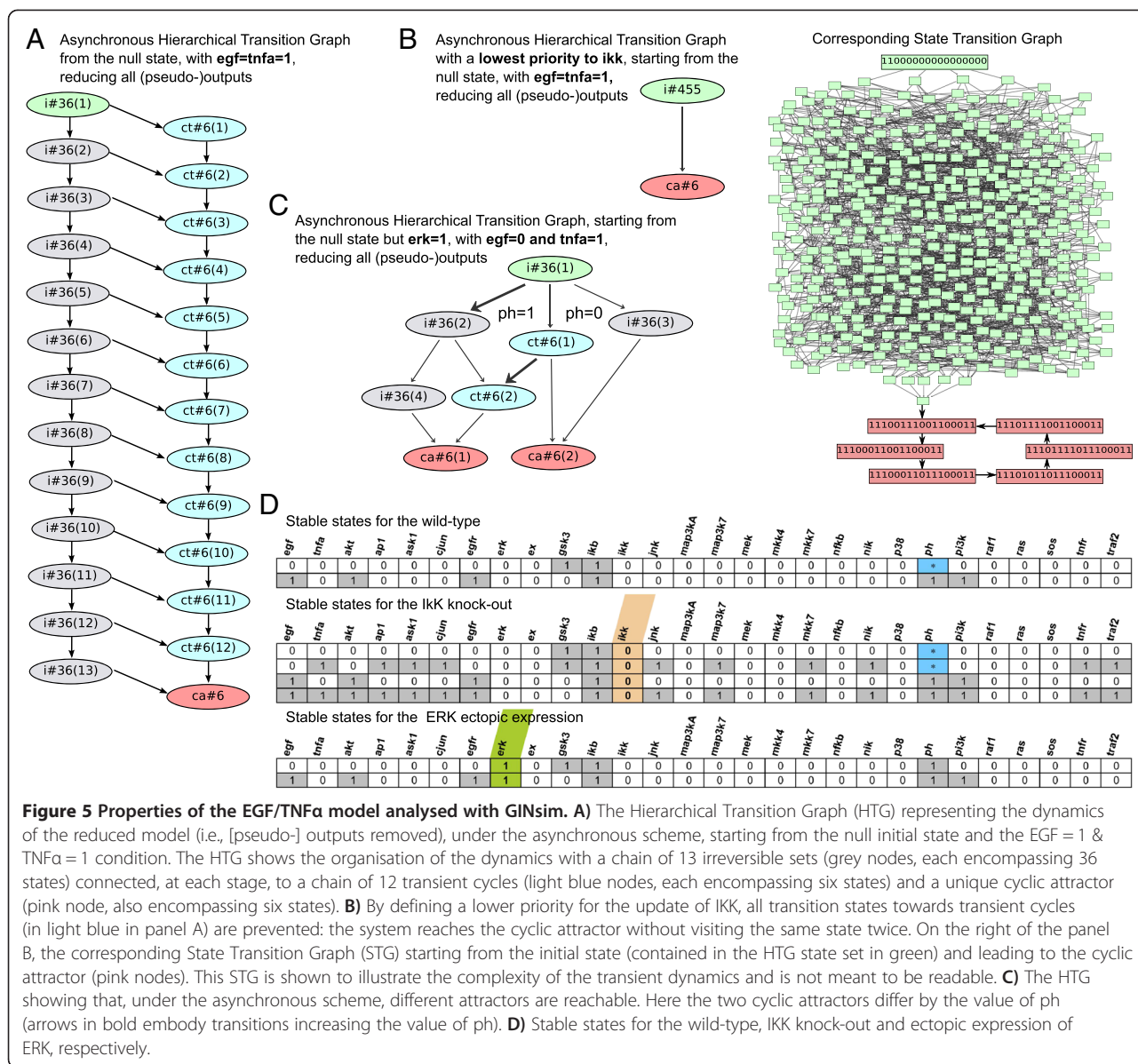simulations, as well as the characteristics of the transient dynamics.

For example, when EGF = 1 and TNFα = 1, the asynchronous state transition graph is substantially larger with about 116 k states, as opposed to the 19 states obtained with a synchronous update (Figure 4B). To contain the size of the state transition graph, we can reduce the model by removing all (pseudo-) output nodes (cf. Methods). Applying this reduction and thereby eliminating 11 nodes as (pseudo-) outputs results in a significant reduction of the state transition graph (down to 546 states, Figure 5A). Importantly, the unique reachable attractor is identical to that obtained with the synchronous update. The resulting HTG (Figure 5A) has a peculiar staged structure with a series of irreversible components (labeled i# followed by the number of states included) that the system may leave to undergo transient oscillations (labeled #ct), to eventually reach the final cyclic attractor (labelled #ca). Interestingly, by defining priority classes and imposing that IKK update is slower than all other species, one can get rid of all these transient oscillations (Figure 5B).

To maintain input components (EGF and TNFα) constant, implicit self-activations are defined. These two
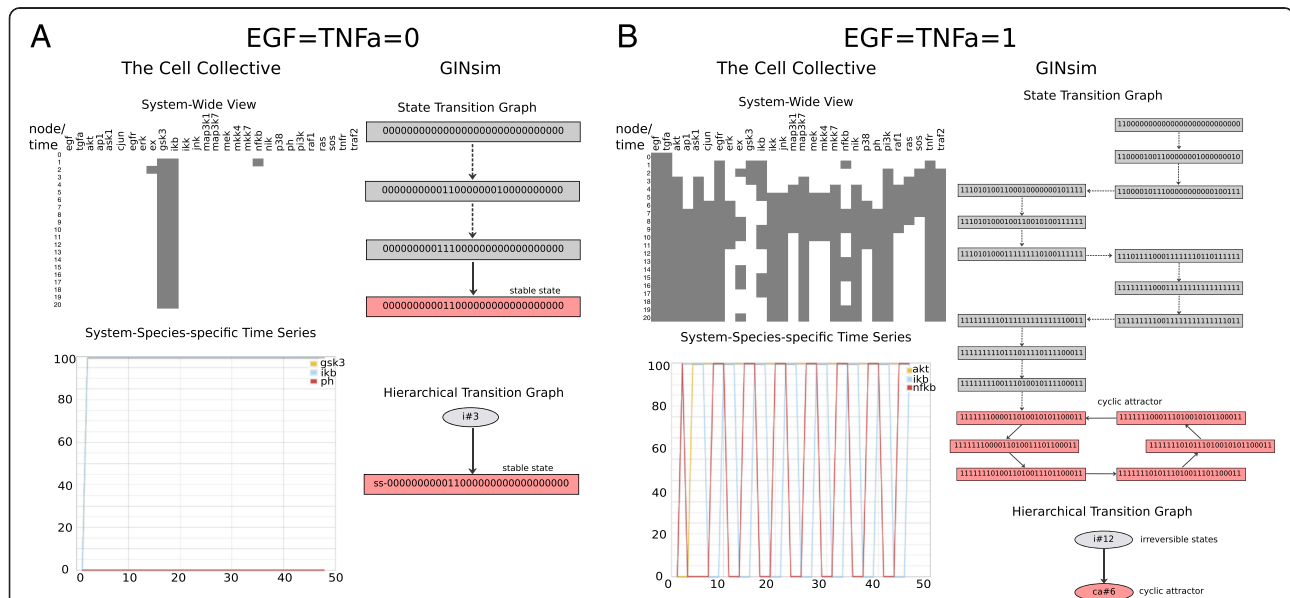
Figure 4 Dynamic profile of the EGF/TNFα model. The model was simulated with consistent results in the Cell Collective (on the left of panels A & B), GINsim (on the right of panels A & B), and CellNOpt (data not shown, but simulations were consistent with those presented here). Synchronous simulations were performed under four input conditions: (i) EGF = TNFα = 0; (ii) EGF = TNFα = 1; (iii) EGF = 0 & TNFα = 1; (iv) EGF = 1 & TNFα = 0. Results for conditions in (i) and (ii) are presented in this figure (the remaining two can be found in Additional file 1). Charts on the top left of each panel correspond to the overall dynamic profile across all nodes in the model. Black cells correspond to active (1) states, whereas inactive (0) states are white. The bottom graphs at the bottom left illustrate the time course of selected nodes. The GINsim columns show State Transition Graph and the Hierarchical Transition Graph (HTG) generated with the tool. Note that due to the synchronous updating, the irreversible components of the (HTG) correspond to linear chains of states. **A)** EGF = TNFα = 0. The network reaches a steady state (shown with both GINsim and the Cell Collective) after 3 transient states. The order of the individual species states in the steady state generated by GINsim is sorted in the same (alphabetical) order, as presented on the Cell Collective side. **B)** EGF = TNFα = 1. After 12 transient states, the network reaches a cyclical attractor encompassing six states. Note that in order to simulate the example model in the Cell Collective as a traditional Boolean network (i.e., with binary input/output), the external species were set to 100 or 0, and the sliding window was set to 1 (see Methods).

Figure 5 Properties of the EGF/TNFα model analysed with GINsim. A) The Hierarchical Transition Graph (HTG) representing the dynamics of the reduced model (i.e., [pseudo-] outputs removed), under the asynchronous scheme, starting from the null initial state and the EGF = 1 & TNFα = 1 condition. The HTG shows the organisation of the dynamics with a chain of 13 irreversible sets (grey nodes, each encompassing 36 states) connected, at each stage, to a chain of 12 transient cycles (light blue nodes, each encompassing six states) and a unique cyclic attractor (pink node, also encompassing six states). **B)** By defining a lower priority for the update of IKK, all transition states towards transient cycles (in light blue in panel A) are prevented: the system reaches the cyclic attractor without visiting the same state twice. On the right of the panel B, the corresponding State Transition Graph (STG) starting from the initial state (contained in the HTG state set in green) and leading to the cyclic attractor (pink nodes). This STG is shown to illustrate the complexity of the transient dynamics and is not meant to be readable. **C)** The HTG showing that, under the asynchronous scheme, different attractors are reachable. Here the two cyclic attractors differ by the value of ph (arrows in bold embody transitions increasing the value of ph). **D)** Stable states for the wild-type, IKK knock-out and ectopic expression of ERK, respectively.

(functional) positive circuits explain the presence of at least four attractors; the combinations of input values define a partition of the state space in four disconnected regions. Using GINsim, we can verify that the EGF/TNFα model encompasses two additional functional circuits: a three-element negative circuit involving IκB, NFκB and ex, and a positive auto-regulatory circuit on ph. The functionality context of the negative circuit corresponds to IKK = 1 (which is the case when TNFα is 1). This negative circuit enables the attractors where IκB, NFκB and ex oscillate. The functionality context of the positive auto-regulatory circuit is defined by ERK = 0. This circuit explains the presence of two attractors when EFG = 0.

For the input configuration where EGF = 0 and TNFα = 1, starting from an initial state with ERK = 1, under the asynchronous update, the system is able to reach two cyclic attractors differing by the presence of ph (Figure 5C). Trajectories leading to the terminal cycle with ph = 0 are discarded by the synchronous update in which the decrease of ERK (in the absence of EGF and thus MEK1) occurs together with the increase of ph, already in the first step of the simulation, leading to the cyclic attractor where ph = 1.

Using GINsim, common perturbations such as gene knock-outs or ectopic gene expressions, as well as their effects, can be easily simulated. For instance, knocking out IKK eliminates oscillations (that were present under wild-type simulations when TNFα = 1), as a result of the interruption of the IκB-NFκB-ex negative circuit (Figure 5D). Similarly, the simulation of ERK ectopic
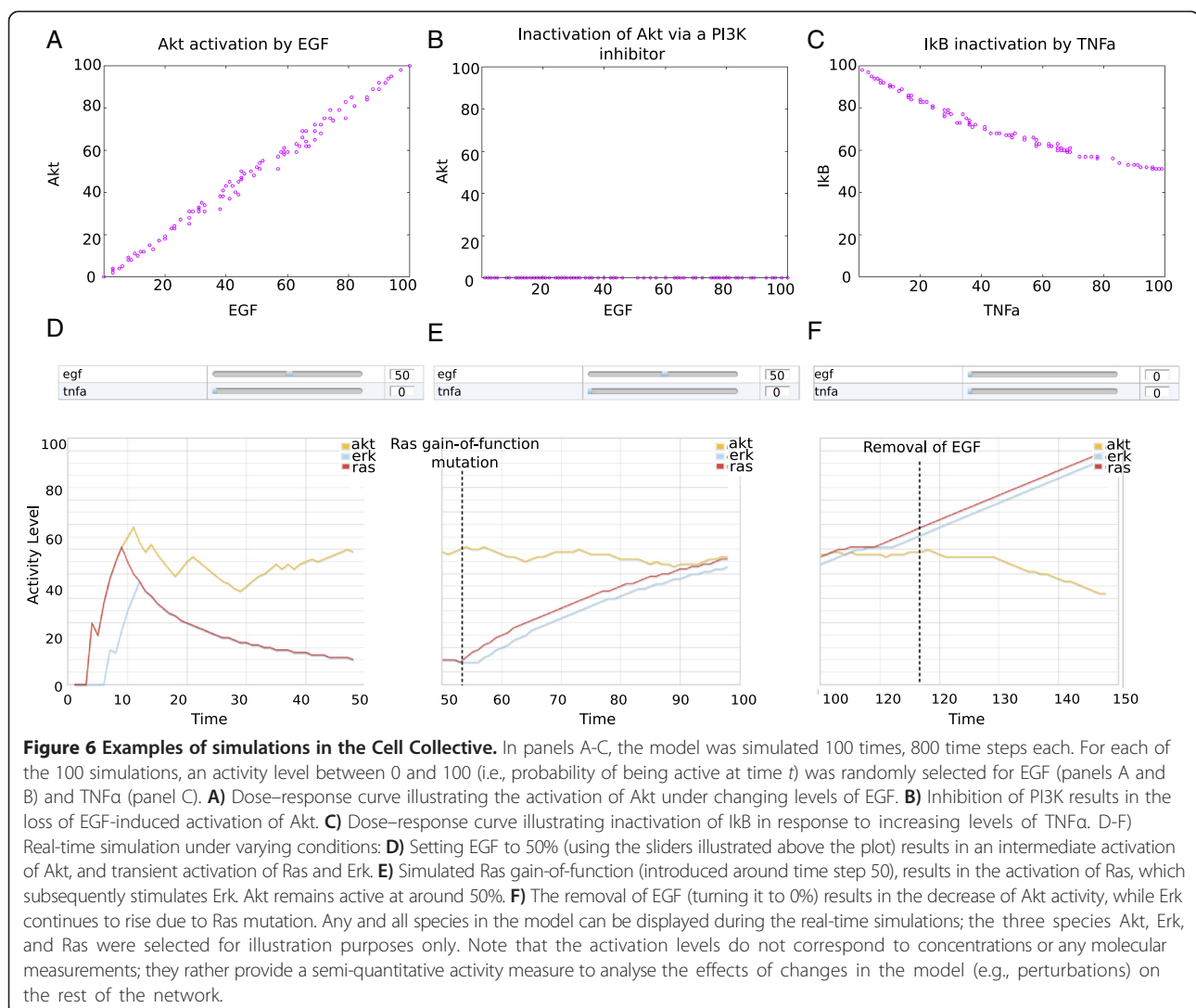
expression, disrupting the functionality context of ph auto-regulation, leads to the loss of the multi-stability when EGF = 0 (Figure 5C).

Simulations and biological application with the cell collective

The Cell Collective platform aims at facilitating collaborative modelling for experimental scientists. The platform offers two modes of simulations: input–output dynamical analyses across hundreds of simulated environments, along with real-time, interactive simulations. Examples of simulations of the EGF/TNFα model with the Cell Collective are illustrated in Figure 6.

Figure 6A-C illustrates simulations of input–output relationships between external species stimulating the network and species of interest regulated in response to this stimulation. Laboratory studies to identify functional relationships between extracellular stimuli and various

cellular components are often expensive and resource consuming. The dynamical analysis tool implemented in the Cell Collective allows users to conduct *in silico* experiments mimicking laboratory experiments, with the advantage that researchers can simulate hundreds or thousands of extracellular and/or disease-related situations (as opposed to the limited number of scenarios that can be reasonably handled in the laboratory) and generate rich input–output relationships (i.e., dose–response curves) between network stimuli and any species in the network. In this respect, inputs and outputs can take continuous values on a scale from 0 to 100 (see Methods), despite the discrete (Boolean) nature of the network model. For example, Figure 6A shows a dose–response curve and a positive correlation between EGF and Akt. In contrast, inhibition of PI3K results in the loss of EGF-dependent activation of Akt (Figure 6B). Finally, the input–output relationship between TNFα and IκB is illustrated in Figure 6C.

Figure 6 Examples of simulations in the Cell Collective. In panels A-C, the model was simulated 100 times, 800 time steps each. For each of the 100 simulations, an activity level between 0 and 100 (i.e., probability of being active at time *t*) was randomly selected for EGF (panels A and B) and TNFα (panel C). **A)** Dose–response curve illustrating the activation of Akt under changing levels of EGF. **B)** Inhibition of PI3K results in the loss of EGF-induced activation of Akt. **C)** Dose–response curve illustrating inactivation of IκB in response to increasing levels of TNFα. D-F) Real-time simulation under varying conditions: **D)** Setting EGF to 50% (using the sliders illustrated above the plot) results in an intermediate activation of Akt, and transient activation of Ras and Erk. **E)** Simulated Ras gain-of-function (introduced around time step 50), results in the activation of Ras, which subsequently stimulates Erk. Akt remains active at around 50%. **F)** The removal of EGF (turning it to 0%) results in the decrease of Akt activity, while Erk continues to rise due to Ras mutation. Any and all species in the model can be displayed during the real-time simulations; the three species Akt, Erk, and Ras were selected for illustration purposes only. Note that the activation levels do not correspond to concentrations or any molecular measurements; they rather provide a semi-quantitative activity measure to analyse the effects of changes in the model (e.g., perturbations) on the rest of the network.

Real-time interactive simulations with the Cell Collective enable users to interactively change the environment during the simulations. This tool enables users to test "what-if" scenarios, e.g., changes in the external conditions, as well as of (transient) gain/loss-of-function, with instant feedback in terms of the changing activity levels of affected species. To illustrate the utility of this mode, we simulated the EGF/TNFα model under a condition where EGF was set to a medium activity level, while keeping TNFα inactive. This condition results in the activation of Akt, Erk, and Ras (Figure 6D). The simulation of a Ras gain-of-function results in further activation of Erk, but not of Akt (Figure 6E). In contrast, Akt continues to respond to EGF activation and deactivation (Figure 6F and G, respectively). This is because Akt (unlike Erk) does not lie downstream of Ras (Figure 2) and hence is not affected by the constitutively activated Ras.

The logical model library

In order to ease the adoption of the new standard, an open source (Java) library, LogicalModel has been created. The library can be used as a standalone command line tool for model conversion, and can be accessed at https://github.com/colomoto/logicalmodel. It provides a data structure to manipulate logical models, as well as a set of analytic tools (e.g., stable state identification, model reduction) that are common to many scientific efforts relying upon a discrete modelling approach. The library further provides import and export filters for SBML qual (through JSBML; see Methods section), as well as an interface enabling the integration of logical models and SBML qual with additional formats and a number of existing software tools. The development of this library coincides with the onset of the CoLoMoTo initiative.

Figure 7 illustrates the central role of the LogicalModel library and main current model exchange capabilities of popular software tools, covering logical models as well as additional related qualitative modelling frameworks. Included are importers that have been recently developed to generate (non parameterised) qualitative models from pathway databases: KEGGtranslator [73] and Path2Models project [74].

Conclusions

In order to enable the interoperability of qualitative, discrete models, a standard exchange format was necessary. While the previous versions of SBML were not fully compatible with qualitative modelling approaches, the modular structure of SBML release (Level 3) enables the development of additional packages to support novel modelling frameworks and capabilities.

In this paper, we report on the SBML Level 3 Qualitative Model (SBML qual) package, which provides a standard means for the exchange of logical models of regulatory and signalling networks. Currently, at least three software tools support both import and export via SBML qual (GINsim, CellNOpt, and the Cell Collective), while other tools such as GNA [75] and CellNetAnalyzer can export models to this format. The former three tools have been used here to demonstrate the consistency of the standard via simulations and analyses of a Boolean model of EGF/TNFα signal transduction pathways. The combined use of software tools is now facilitated, providing modellers with a range of complementary means to investigate their models.

Repositories of models encoded in SBML qual are already being prepared. For instance, the Cell Collective now contains numerous previously published logical models that can be downloaded. BioModels Database is a repository of computational models of biological networks

Figure 7 LogicalModel library as an interface among various qualitative modelling technologies. Orange boxes denote formats, light yellow boxes denote software tools and white boxes correspond to libraries. Arrows denote export/import capabilities.
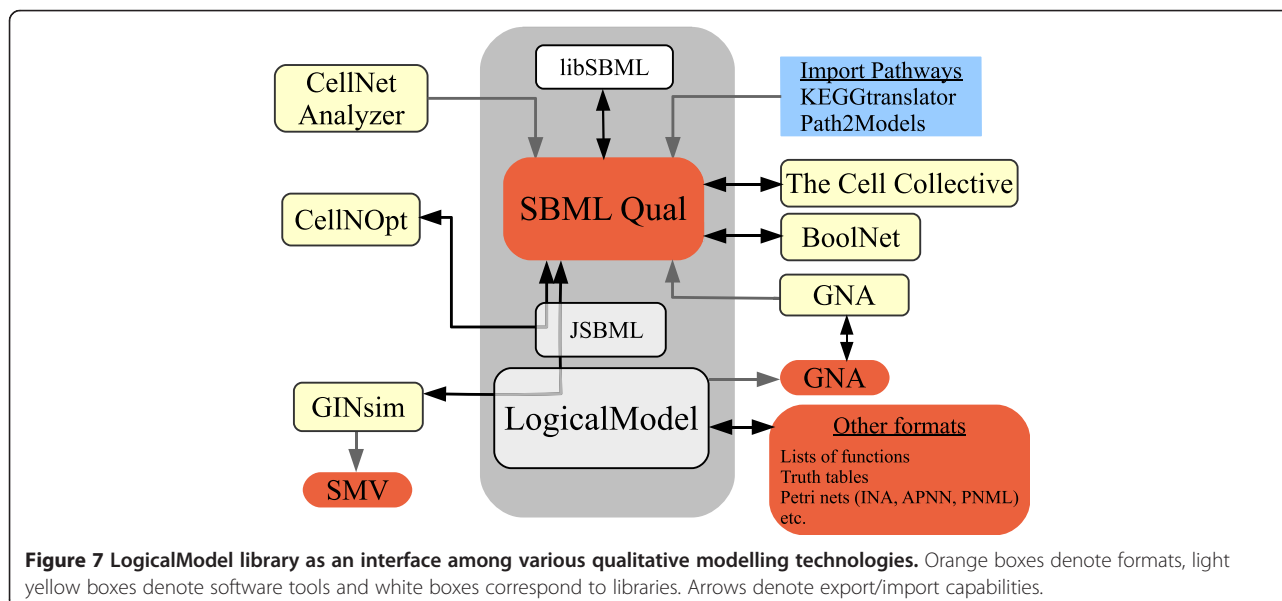
[76] that currently hosts several curated SBML qual models (http://www.ebi.ac.uk/biomodels/).

SBML qual will continue to be refined by the community. Some of the improvements discussed so far include, for example, the definition of models where parameters are not (all) instantiated, models for which timing constraints (or rates) are specified, extended Petri nets, etc. In addition, further integration with SBML Core concepts is planned. In particular, SBML qual will be useful to so-called hybrid formalisms, which combine features of both discrete and continuous models. A typical example are formalisms embedding a logical representation of the interaction structure of the network into a continuous model of its dynamics, such as piecewise-linear differential equation models [77], hybrid automata [78,79], or even fully continuous models in which the logical functions have been replaced by sigmoidal functions preserving the logic of the interactions [80]. Other hybrid formalisms that have been used for the modelling of regulatory and signalling networks are fuzzy logic-based models [81] and timed automata [82,83]. Software tools enabling the modelling, simulation, and analysis of networks by means of different kinds of hybrid models include CellNOpt, Odefy [39], SQUAD [41], GNA [75], and Q2LM [84]. Most of the these tools support SBML Core.

Last but not least, to further support data and result reproducibility, the standardisation of algorithms and simulation schemes and parameters for qualitative models is planned by adopting the MIASE (Minimum Information About a Simulation Experiment) guidelines [85]. First steps in this direction have already been taken by adding simulation algorithms relevant to logical models to the Kinetic Simulation Algorithm Ontology (KiSAO [86]), allowing qualitative models to be used in simulations described in the Simulation Experiment Description Markup Language (SED-ML [87]).

The availability of SBML qual and the inception of the CoLoMoTo consortium should foster the collaborative development of standards (including the extension of existing ones), as well as of computational methods for the qualitative modelling of biological networks. In this respect, anyone interested in these efforts is cordially invited to enter into contact with the existing community at sbml-qual@lists.sourceforge.net.

Additional file

Additional file 1: Complete dynamic profile of the example model.
The model was simulated with consistent results in the Cell Collective (left column of panels A & B), GINsim (right column of panels A & B), and CellNOpt (data not shown, but simulations were consistent with those presented here). Charts at the top of the Cell Collective column correspond to the overall dynamic profile across all nodes in the model. Black cells correspond to active (1) states, whereas inactive (0) states are white. The bottom graphs in the Cell Collective column illustrate the time course of selected nodes. The GINsim columns show State Transition Graph and the Hierarchical Transition Graph (HTG) generated with the tool. Note that due to the synchronous updating, the irreversible components of the (HTG) correspond to linear chains of states. Simulations were performed under four input conditions: EGF = TNFα = 0; EGF = TNFα = 1; EGF = 0 & TNFα = 1; EGF = 1 & TNFα = 0. A) EGF = TNFa = 0. The network reaches a steady state (shown in both GINsim and the Cell Collective column) after 3 transient states. The order of the individual species states in the steady state generated by GINsim is sorted in the same (alphabetical) order, as presented in the Cell Collective column. B) EGF = 1, TNFa = 0. The network reaches a steady state after 14 transient states. C) EGF = 0, TNFa = 1. Following 5 transient states, the network reaches a 6-cycle attractor. D) EGF = TNFa = 1. After 12 transient states, the network reaches a cyclical attractor encompassing six states.

Author details

[1]Instituto Gulbenkian de Ciência, Rua da Quinta Grande 6, 2780-156 Oeiras, Portugal. [2]Institut de Mathématiques de Luminy, Campus de Luminy, Case 907, 13288 Marseille Cedex 9, France. [3]European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK. [4]Center for Integrative Genomics, University of Lausanne, CH-1015 Lausanne, Switzerland. [5]The Babraham Institute, Babraham Research Campus, Cambridge CB22 3AT, UK. [6]Department of Bioengineering, University of California San Diego, La Jolla, CA 92093-0412, USA. [7]Center for Bioinformatics Tuebingen (ZBIT), University of Tuebingen, 72076 Tübingen, Germany. [8]College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182, USA. [9]Swiss-Prot & Vital-IT group, SIB- Swiss Institute of Bioinformatics, Center for Integrative Genomics, University of Lausanne, Quartier Sorge - Batiment Genopode, CH-1015 Lausanne, Switzerland. [10]INRIA Grenoble – Rhône-Alpes, 655 avenue de l'Europe, Montbonnot, 38334 Saint-Ismier Cedex, France. [11]IAE Grenoble, Université Pierre-Mendès-France, Domaine universitaire BP 47, 38040 Grenoble Cedex 9, France. [12]Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID), Rua Alves Redol 9, 1000-029 Lisbon, Portugal. [13]Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, D-39106 Magdeburg, Germany. [14]Computing and Mathematical sciences, California Institute of Technology, Pasadena, CA 91125, USA. [15]Institut de Biologie de l'Ecole Normale Supérieure (IBENS) - UMR CNRS 8197 - INSERM 1024 46 rue d'Ulm, 75230 Paris Cedex 05, France. [16]Department of Biochemistry, University of Nebraska-Lincoln, Lincoln, NE 68588, USA.

References

1. Kauffman S: **Homeostasis and differentiation in random genetic control networks.** *Nature* 1969, **224**:177–8.
2. Thomas R: **Boolean formalisation of genetic control circuits.** *J Theor Biol* 1973, **42**:565–583.
3. Silva-Rocha R, de Lorenzo V: **The TOL network of Pseudomonas putida mt-2 processes multiple environmental inputs into a narrow response space.** *Environ Microbiol* 2013, **15**:271–86.
4. Herrmann F, Groß A, Zhou D, Kestler HA, Kühl M: **A boolean model of the cardiac gene regulatory network determining first and second heart field identity.** *PLoS One* 2012, **7**:e46798.
5. Akman OE, Watterson S, Parton A, Binns N, Millar AJ, Ghazal P: **Digital clocks: simple Boolean models can quantitatively describe circadian systems.** *J R Soc Interface* 2012, **9**:2365–82.
6. Giacomantonio CE, Goodhill GJ: **A Boolean model of the gene regulatory network underlying mammalian cortical area development.** *PLoS Comput Biol* 2010, **6**:e1000936.
7. Tokar T, Turcan Z, Ulicny J: **Boolean network-based model of the Bcl-2 family mediated MOMP regulation.** *Theor Biol Med Model* 2013, **10**:40.
8. Sridharan S, Layek R, Datta A, Venkatraj J: **Boolean modeling and fault diagnosis in oxidative stress response.** *BMC Genomics* 2012, **13**(Suppl 6):S4.
9. Domedel-Puig N, Rué P, Pons AJ, García-Ojalvo J: **Information routing driven by background chatter in a signaling network.** *PLoS Comput Biol* 2011, **7**:e1002297.
10. Saadatpour A, Albert I, Albert R: **Attractor analysis of asynchronous Boolean models of signal transduction networks.** *J Theor Biol* 2010, **266**:641–56.
11. Naldi A, Carneiro J, Chaouiya C, Thieffry D: **Diversity and plasticity of Th cell types predicted from regulatory network modelling.** *PLoS Comput Biol* 2010, **6**:e1000912.
12. Samaga R, Saez-Rodriguez J, Alexopoulos LG, Sorger PK, Klamt S: **The logic of EGFR/ErbB signaling: theoretical properties and analysis of high-throughput data.** *PLoS Comput Biol* 2009, **5**:e1000438.
13. Calzone L, Tournier L, Fourquet S, Thieffry D, Zhivotovsky B, Barillot E, Zinovyev A: **Mathematical modelling of cell-fate decision in response to death receptor engagement.** *PLoS Comput Biol* 2010, **6**:e1000702.
14. Helikar T, Konvalina J, Heidel J, Rogers JA: **Emergent decision-making in biological signal transduction networks.** *Proc Natl Acad Sci U S A* 2008, **105**:1913–1918.
15. Fauré A, Naldi A, Chaouiya C, Thieffry D: **Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle.** *Bioinformatics* 2006, **22**:e124–31.
16. Davidich MI, Bornholdt S: **Boolean network model predicts cell cycle sequence of fission yeast.** *PLoS One* 2008, **3**:e1672.
17. Todd RG, Helikar T: **Ergodic sets as cell phenotype of budding yeast cell cycle.** *PLoS One* 2012, **7**:e45780.
18. Sahin O, Fröhlich H, Löbke C, Korf U, Burmester S, Majety M, Mattern J, Schupp I, Chaouiya C, Thieffry D, Poustka A, Wiemann S, Beissbarth T, Arlt D: **Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance.** *BMC Syst Biol* 2009, **3**:1.
19. Madrahimov A, Helikar T, Kowal B, Lu G, Rogers J: **Dynamics of influenza virus and human host interactions during infection and replication cycle.** *Bull Math Biol* 2013, **75**:988–1011.
20. Feuer R, Gottlieb K, Viertel G, Klotz J, Schober S, Bossert M, Sawodny O, Sprenger G, Ederer M: **Model-based analysis of an adaptive evolution experiment with Escherichia coli in a pyruvate limited continuous culture with glycerol.** *EURASIP J Bioinform Syst Biol* 2012, **2012**:14.
21. Kazemzadeh L, Cvijovic M, Petranovic D: **Boolean model of yeast apoptosis as a tool to study yeast and human apoptotic regulations.** *Front Physiol* 2012, **3**:446.
22. Li F, Long T, Lu Y, Ouyang Q, Tang C: **The yeast cell-cycle network is robustly designed.** *Proc Natl Acad Sci U S A* 2004, **101**:4781–6.
23. Fauré A, Naldi A, Lopez F, Chaouiya C, Ciliberto A, Thieffry D: **Modular logical modelling of the budding yeast cell cycle.** *Mol Biosyst* 2009, **5**:1787–96.
24. Albert R, Othmer HG: **The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster.** *J Theor Biol* 2003, **223**:1–18.
25. Sánchez L, Chaouiya C, Thieffry D: **Segmenting the fly embryo: logical analysis of the role of the segment polarity cross-regulatory module.** *Int J Dev Biol* 2008, **52**:1059–75.
26. González A, Chaouiya C, Thieffry D: **Logical modelling of the role of the Hh pathway in the patterning of the Drosophila wing disc.** *Bioinformatics* 2008, **24**:i234–40.
27. Li S, Assmann SM, Albert R: **Predicting essential components of signal transduction networks: a dynamic model of guard cell abscisic acid signaling.** *PLoS Biol* 2006, **4**:e312.
28. Mendoza L, Thieffry D, Alvarez-Buylla ER: **Genetic control of flower morphogenesis in Arabidopsis thaliana: a logical analysis.** *Bioinformatics* 1999, **15**:593–606.
29. Helikar T, Kochi N, Kowal B, Dimri M, Naramura M, Raja SM, Band V, Band H, Rogers JA: **A comprehensive, multi-scale dynamical model of ErbB receptor signal transduction in human mammary epithelial cells.** *PLoS One* 2013, **8**:e61757.
30. Terfve CDA, Cokelaer T, Henriques D, Macnamara A, Gonçalves E, Morris MK, van Iersel M, Lauffenburger DA, Saez-Rodriguez J: **CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms.** *BMC Syst Biol* 2012, **6**:133.
31. Hinkelmann F, Brandon M, Guang B, McNeill R, Blekherman G, Veliz-Cuba A, Laubenbacher R: **ADAM: analysis of discrete models of biological systems using computer algebra.** *BMC Bioinforma* 2011, **12**:295.
32. Müssel C, Hopfensitz M, Kestler HA: **BoolNet–an R package for generation, reconstruction and analysis of Boolean networks.** *Bioinformatics* 2010, **26**:1378–80.
33. Albert I, Thakar J, Li S, Zhang R, Albert R: **Boolean network simulations for life scientists.** *Source Code Biol Med* 2008, **3**:16.
34. Helikar T, Kowal B, Rogers JA: **A cell simulator platform: the cell collective.** *Clin Pharmacol Ther* 2013, **93**:393–5.
35. Helikar T, Kowal B, McClenathan B, Bruckner M, Rowley T, Wicks B, Shrestha M, Limbu K, Rogers JA: **The cell collective: toward an open and collaborative approach to systems biology.** *BMC Syst Biol* 2012, **6**:96.
36. Klamt S, Saez-Rodriguez J, Gilles ED: **Structural and functional analysis of cellular networks with cell NetAnalyzer.** *BMC Syst Biol* 2007, **1**:2.

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

38. Naldi A, Berenguier D, Fauré A, Lopez F, Thieffry D, Chaouiya C: **Logical modelling of regulatory networks with GINsim 2.3.** *Biosystems* 2009, **97**:134–139.

39. Krumsiek J, Pölsterl S, Wittmann DM, Theis FJ: **Odefy–from discrete to continuous models.** *BMC Bioinform* 2010, **11**:233.

40. Zheng J, Zhang D, Przytycki PF, Zielinski R, Capala J, Przytycka TM: **SimBoolNet–a cytoscape plugin for dynamic simulation of signaling networks.** *Bioinformatics* 2010, **26**:141–2.

41. Di Cara A, Garg A, De Micheli G, Xenarios I, Mendoza L: **Dynamic simulation of regulatory networks using SQUAD.** *BMC Bioinform* 2007, **8**:462.

42. Clarke EM, Grumberg O, Peled D: *Model-checking.* Cambridge, MA: MIT Press; 1999:330.

43. Batt G, Ropers D, de Jong H, Geiselmann J, Mateescu R, Page M, Schneider D: **Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in Escherichia coli.** *Bioinformatics* 2005, **21**(Suppl 1):i19–28.

44. Cimatti A, Clarke E, Giunchiglia E, Giunchiglia F, Pistore M, Roveri M, Sebastiani R, Tacchella A: **NuSMV 2: an OpenSource tool for symbolic model checking.** *Lect Notes Comp Sci* 2002, **2404**:359–64.

45. Bernot G, Comet J-P, Richard A, Guespin J: **Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic.** *J Theor Biol* 2004, **229**:339–47.

46. Chabrier-Rivier N, Fages F, Soliman S: **The biochemical abstract machine BIOCHAM.** *Lect Notes Comp Sci* 2005, **3082**:171–91.

47. Monteiro PT, Chaouiya C: **Efficient verification for logical models of regulatory networks.** *Adv Intell Soft Comput* 2012, **154**:259–67.

48. Monteiro PT, Ropers D, Mateescu R, Freitas AT, de Jong H: **Temporal logic patterns for querying dynamic models of cellular interaction networks.** *Bioinformatics* 2008, **24**:i227–33.

49. Carrillo M, Góngora PA, Rosenblueth DA: **An overview of existing modeling tools making use of model checking in the analysis of biochemical networks.** *Front Plant Sci* 2012, **3**:155.

50. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr J-H, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, *et al*: **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models.** *Bioinformatics* 2003, **19**:524–531.

51. Le Novère N, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, Demir E, Wegner K, Aladjem MI, Wimalaratne SM, Bergman FT, Gauges R, Ghazal P, Kawaji H, Li L, Matsuoka Y, Villéger A, Boyd SE, Calzone L, Courtot M, Dogrusoz U, Freeman TC, Funahashi A, Ghosh S, Jouraku A, Kim S, Kolpakov F, Luna A, Sahle S, Schmidt E, *et al*: **The systems biology graphical notation.** *Nat Biotechnol* 2009, **27**:735–41.

52. Hardy S, Robillard PN: **Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches.** *J Bioinform Comput Biol* 2004, **2**:595–613.

53. Chaouiya C: **Petri net modelling of biological networks.** *Brief Bioinform* 2007, **8**:210–9.

54. Koch I, Heiner M: **Petri nets.** In *Biol Netw Anal.* Edited by Junker BH, Schreiber F. Hoboken, New Jersey: John Wiley & Sons; 2008:139–179.

55. Rohr C, Marwan W, Heiner M: **Snoopy–a unifying Petri net framework to investigate biomolecular networks.** *Bioinformatics* 2010, **26**:974–5.

56. Bornstein BJ, Keating SM, Jouraku A, Hucka M: **LibSBML: an API library for SBML.** *Bioinformatics* 2008, **24**:880–1.

57. Dräger A, Rodriguez N, Dumousseau M, Dörr A, Wrzodek C, Le Novère N, Zell A, Hucka M: **JSBML: a flexible Java library for working with SBML.** *Bioinformatics* 2011, **27**:2167–8.

58. Saez-Rodriguez J, Alexopoulos LG, Epperlein J, Samaga R, Lauffenburger DA, Klamt S, Sorger PK: **Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction.** *Mol Syst Biol* 2009, **5**:331.

59. Klamt S, Saez-Rodriguez J, Lindquist JA, Simeoni L, Gilles ED: **A methodology for the structural and functional analysis of signaling and regulatory networks.** *BMC Bioinform* 2006, **7**:56.

60. Egea JA, Henriques D, Cokelaer T, Villaverde AF, Banga JR, Saez-Rodriguez J: *MEIGO: a software suite based on metaheuristics for global optimization in systems biology and bioinformatics.* Submitted (available on arXiv.org: http://arxiv.org/abs/1311.5735).

61. Guziolowski C, Videla S, Eduati F, Thiele S, Cokelaer T, Siegel A, Saez-Rodriguez J: **Exhaustively characterizing feasible logic models of a signaling network using answer Set programming.** *Bioinformatics* 2013, **29**:2320–6.

62. Saez-Rodriguez J, Alexopoulos LG, Zhang M, Morris MK, Lauffenburger DA, Sorger PK: **Comparing signaling networks between normal and transformed hepatocytes using discrete logical models.** *Cancer Res* 2011, **71**:5400–11.

63. Eduati F, De Las Rivas J, Di Camillo B, Toffolo G, Saez-Rodriguez J: **Integrating literature-constrained and data-driven inference of signalling networks.** *Bioinformatics* 2012, **28**:2311–2317.

64. Chaouiya C, Naldi A, Thieffry D: **Logical modelling of gene regulatory networks with GINsim.** *Methods Mol Biol* 2012, **804**:463–79.

65. Thieffry D: **Dynamical roles of biological regulatory circuits.** *Brief Bioinform* 2007, **8**:220–5.

66. Bérenguier D, Chaouiya C, Monteiro PT, Naldi A, Remy E, Thieffry D, Tichit L: **Dynamical modeling and analysis of large cellular regulatory networks.** *Chaos An Interdisc J Nonlinear Sci* 2013, **23**:025114.

67. Naldi A, Remy E, Thieffry D, Chaouiya C: **Dynamically consistent reduction of logical regulatory graphs.** *Theor Comput Sci* 2011, **412**(21):2207–18.

68. Naldi A, Monteiro P, Chaouiya C: **Efficient handling of large signalling-regulatory networks by focusing on their core control.** *Lect Notes Comp Sci* 2012, **7605**:288–306.

69. Helikar T, Kowal B, Madrahimov A, Shrestha M, Pedersen J, Limbu K, Thapa I, Rowley T, Satalkar R, Kochi N, Konvalina J, Rogers JA: **Bio-logic builder: a non-technical tool for building dynamical, qualitative models.** *PLoS One* 2012, **7**:e46417.

70. *Qualitative models, version 1 release 1.* http://identifiers.org/combine.specifications/sbml.level-3.version-1.qual.version-1.release-1.

71. MacNamara A, Terfve C, Henriques D, Bernabé BP, Saez-Rodriguez J: **State-time spectrum of signal transduction logic models.** *Phys Biol* 2012, **9**:045003.

72. Gonçalves E, van Iersel M, Saez-Rodriguez J: **CySBGN: a cytoscape plug-in to integrate SBGN maps.** *BMC Bioinform* 2013, **14**:17.

73. Wrzodek C, Büchel F, Ruff M, Dräger A, Zell A: **Precise generation of systems biology models from KEGG pathways.** *BMC Syst Biol* 2013, **7**:15.

74. Büchel F, Rodriguez N, Swainston N, Wrzodek C, Czauderna T, Keller R, Mittag F, Schubert M, Glont M, Golebiewski M, van Iersel M, Keating S, Rall M, Wybrow M, Hermjakob H, Hucka M, Kell DB, Müller W, Mendes P, Zell A, Chaouiya C, Saez-Rodriguez J, Schreiber F, Laibe C, Dräger A, Le Novère N: **Path2Models: large-scale generation of computational models from biochemical pathway maps.** *BMC Syst Biol* 2013, **7**:116.

75. Batt G, Besson B, Ciron PE, de Jong H, Dumas E, Geiselmann J, Monte R, Monteiro PT, Page M, Rechenmann F, Ropers D: **Genetic network analyzer: a tool for the qualitative modeling and simulation of bacterial regulatory networks.** *Methods Mol Biol* 2012, **804**:439–62.

76. Chelliah V, Laibe C, Le Novère N: **BioModels database: a repository of mathematical models of biological processes.** *Methods Mol Biol* 2013, **1021**:189–99.

77. Glass L, Kauffman SA: **The logical analysis of continuous, non-linear biochemical control networks.** *J Theor Biol* 1973, **39**:103–29.

78. Alur R, Belta C, Ivančić F, Kumar V, Mintz M, Pappas GJ, Rubin H, Schug J: *Hybrid modeling and simulation of biomolecular network,* Hybrid Syst Comput Control. Heidelberg: Springer Berlin Heidelberg; 2001:19–32.

79. Brazma A, Cerans K, Ruklisa D, Schlitt T, Viksna J: **HSM - a hybrid system based approach for modelling intracellular networks.** *Gene* 2013, **518**:70–7.

80. Mestl T, Plahte E, Omholt SW: **A mathematical framework for describing and analysing gene regulatory networks.** *J Theor Biol* 1995, **176**:291–300.

81. Aldridge BB, Saez-Rodriguez J, Muhlich JL, Sorger PK, Lauffenburger DA: **Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/insulin-induced signaling.** *PLoS Comput Biol* 2009, **5**:e1000340.

82. Batt G, Salah RB, Maler O: **On timed models of gene networks.** In *Form model anal timed syst: volume 4763.* Edited by Raskin J-F, Thiagarajan PS. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007:38–52.

83. Siebert H, Bockmayr A: **Temporal constraints in the logical analysis of regulatory networks.** *Theor Comput Sci* 2008, **391**:258–275.

84. Morris MK, Shriver Z, Sasisekharan R, Lauffenburger DA: **Querying quantitative logic models (Q2LM) to study intracellular signaling networks and cell-cytokine interactions.** *Biotechnol J* 2012, **7**:374–86.

85. Waltemath D, Adams R, Beard DA, Bergmann FT, Bhalla US, Britten R, Chelliah V, Cooling MT, Cooper J, Crampin EJ, Garny A, Hoops S, Hucka M, Hunter P, Klipp E, Laibe C, Miller AK, Moraru I, Nickerson D, Nielsen P, Nikolski M, Sahle S, Sauro HM, Schmidt H, Snoep JL, Tolle D, Wolkenhauer O, Le Novère N: **Minimum information about a simulation experiment (MIASE).** *PLoS Comput Biol* 2011, **7**:e1001122.

86. Courtot M, Juty N, Knüpfer C, Waltemath D, Zhukova A, Dräger A, Dumontier M, Finney A, Golebiewski M, Hastings J, Hoops S, Keating S, Kell DB, Kerrien S, Lawson J, Lister A, Lu J, Machne R, Mendes P, Pocock M, Rodriguez N, Villeger A, Wilkinson DJ, Wimalaratne S, Laibe C, Hucka M, Le Novère N: **Controlled vocabularies and semantics in systems biology.** *Mol Syst Biol* 2011, **7**:543.

87. Waltemath D, Adams R, Bergmann FT, Hucka M, Kolpakov F, Miller AK, Moraru II, Nickerson D, Sahle S, Snoep JL, Le Novère N: **Reproducible computational biology experiments with SED-ML–the simulation experiment description markup language.** *BMC Syst Biol* 2011, **5**:198.