bioRxiv preprint doi: https://doi.org/10.1101/2020.03.22.998377. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

Reconciling Qualitative, Abstract, and Scalable Modeling of Biological Networks

Loïc Paulevé[a,b,✉], **Juraj Kolčák**[c], **Thomas Chatain**[c], **and Stefan Haar**[c]

[a]Université Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, Talence, France
[b]LRI UMR8623, Université Paris-Sud, CNRS, Université Paris-Saclay, France
[c]Inria and LSV, CNRS (UMR 8643) and ENS Paris-Saclay, Université Paris-Saclay, France

Predicting the behaviors of complex biological systems, underpinning processes such as cellular differentiation, requires taking into account many molecular and genetic elements for which limited information is available past a global knowledge of their pairwise interactions. Logical modeling, notably with Boolean Networks (BNs), is a well-established approach which enables reasoning on the qualitative dynamics of networks accounting for many species. Several dynamical approaches have been proposed to interpret the logic of the regulations encoded by the BNs. The synchronous and (fully) asynchronous ones are the most prominent, where the value of either all or only one component can change at each step. Here we prove that, besides being costly to analyze, these usual interpretations are not adequate to represent quantitative systems, being able to both predict spurious behaviors and miss others. We introduce a new paradigm, the Most Permissive Boolean Networks (MPBNs), which offer the formal guarantee not to miss any behavior achievable by a quantitative model following the same logic. Moreover, MPBNs significantly reduce the complexity of dynamical analysis, enabling to model genome-scale networks.

computational modeling | automatic reasoning | simulations | systems biology

Correspondence (✉): loic.pauleve@labri.fr

DRAFT

Models in systems biology typically integrate knowledge and hypotheses on molecular interactions, manually or semi-automatically, gathered from experimental data found in databases and the literature. These models are often qualified as "mechanistic," in opposition to those solely based on biophysical laws.

Since their introduction in the late '60s (1, 2), logical models, such as Boolean Networks, have been widely adopted for reasoning about signaling and gene networks (3–11) as they require few parameters and can easily integrate information from omics datasets and genetic screens. These models represent processes with a high degree of generalization and can offer coarse-grained but robust predictions. That makes them particularly suitable for large biological networks, for which ample global knowledge exists about potential interactions with little precise data on actual molecules abundances and reaction kinetics.

The validation of computational models is necessary to trust their subsequent predictions. In systems biology, validation primarily involves *in silico* reproduction of observed behaviors by executing the computational model. Such observations may be measurements of the activity, over time, or at steady-state, of some of the interacting molecules under different experimental conditions. Therefore, if no executions of a BN reproduce an experimentally observed behavior (e.g., the activation of a particular gene), the model, and the associated interactions, is considered as invalid. This procedure also enables general studies on interaction motifs that are necessary or sufficient for achieving fundamental behaviors such as cellular differentiation or homeostasis (12–15).

Boolean Networks are often created from scratch and rather than derived from a detailed mechanistic (partially-parameterized) model. Consequently, there is no guarantee that their analysis can be relevant for a more precise model, and thus for the actual biological system.

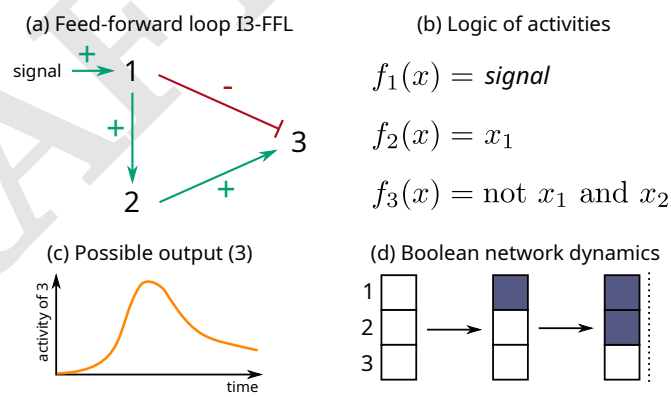

Fig. 1. Incoherent feed-forward loop of type 3 (a) and its associated Boolean logic for nodes activities (b); $f_{1,2,3}(x)$ are the Boolean functions used to compute the next value of each node from a given *configuration* x of the network, which is here a binary vector specifying the current value of each node, x_i referring to the Boolean value of node i. Whereas theoretical and experimental studies show a possible activation of the output when the signal is active (c), usual BNs analysis cannot predict this transient behavior: (d) shows the corresponding complete dynamics of f, where configurations are represented by piles of 3 squares, where the top square represents the state of the first component, and so forth. A white square represents the inactive (0) state; a blue square represents the active (1) state; a dashed line indicates that no further evolution is possible. Arrows indicate possible transitions. The node 3 is never predicted to be active.

Fig. 1 illustrates this issue with the incoherent feed-forward loop of type 3, I3-FFL(16). An input node 1 directly inhibits the output 3, but indirectly activates it via node 2. Theoretical studies and experimental data from synthetically designed circuits show that a monotonic activation of the input can lead to a transient activity of the output. However, it is impossible to reproduce this behavior with usual interpretations of BNs, including synchronous and asynchronous: if 1 is not active, neither 2 nor 3 can be activated. If 1 is active, 2 is active, but any transient activation of 3 is prevented (Fig. 1(d)).

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.22.998377. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

Additional model features, such as intermediate levels for the nodes, or delays in interactions, would allow a transient activation for the I3-FFL output. However, such features come with additional parameters and higher computational cost, which limits their general application to large scale networks.

This simple example seems to show that setting binary activities for nodes can both generate spurious behaviors (as expected with qualitative models), and also preclude the discovery of existing behaviors. Therefore, the validity of a model cannot be assessed by the usual interpretations of BNs. This limitation largely impedes the inference of dynamical network models and the identification of necessary interaction network motifs since the Boolean interpretation can strongly distort the landscape of candidate models.

However, we found that this issue is actually due to the usual interpretations of BNs and not to their intrinsic Boolean nature. We introduce a new simulation approach, the *Most Permissive Boolean Networks* (MPBNs), which presents the formal guarantee to capture all behaviors achievable without the need for additional parameters. If MPBNs cannot reproduce a given observation, no quantitative refinement of the Boolean model can do it, and the model can safely be considered as incoherent with the observations. While predicting more behaviors than the usual interpretations of Boolean Networks, MPBNs still capture essential dynamical features of biological models.

Moreover, we demonstrated that the analysis of MPBNs avoids the state space explosion problem, a strong limiting factor for the usual interpretations of BNs. The drastically reduced computational cost enables the precise qualitative analysis of dynamics of genome-scale networks.

Modeling with Boolean Networks

Computational modeling of dynamical systems relies on two fundamental ingredients: a language to specify the model, and an execution semantics. The language provides symbols and syntax rules to write a model, while the execution semantics mathematically defines how to interpret it. The semantics formalizes the notion of network configurations (or states) and how to compute their evolution over time. It provides an exhaustive assessment of model capabilities by enabling dynamical analyzes such as simulations as well as formal verification by invariant analysis and model-checking.

A BN is specified by a mathematical function mapping any binary vector of dimension n to another binary vector of the same dimension:

$$f : \mathbb{B}^n \to \mathbb{B}^n \qquad [1]$$

where $\mathbb{B} = \{0, 1\}$ represent the Boolean values. Each element of a binary vector models the state (inactive/active, absent/present) of the associated network node, and f_i is the function which specifies the state towards which the i-th element evolves. Fig. 2(b) gives an example of a BN modeling a switch system.

BNs semantics computes the possible temporal evolutions of the component states using different methods. With *synchronous* executions of BNs (introduced by S. Kauffman (1)), we update all the components of the network at the same time, and a configuration $x \in \mathbb{B}^n$ can only evolve to one configuration $f(x)$. With *fully asynchronous* executions of BNs (introduced by R. Thomas and usually referred to more simply as *asynchronous* in the computational systems biology literature), we update only one component at a given time, and a configuration $x \in \mathbb{B}^n$ can evolve to any configuration which differs only by a single component i where $f_i(x) \neq x_i$. This introduces potential *non-determinism* in the model trajectory since there can be different executions of the same BN from a given initial configuration. The (fully) asynchronous semantics is often described as more realistic for modeling biological networks, accounting for different kinetics of interactions.

Many more variants of executions of BNs have been studied in the literature, some imposing a precise order in the updating of the components, others allowing subsets of components to be updated simultaneously, etc. Most, if not all, generate a subset of the executions achievable with the (generalized) *asynchronous* semantics of BNs where any number of components can be updated at a time: a configuration can evolve to any other configuration that complies with the logical functions for the components that differ between both. Formally, for any $x, y \in \mathbb{B}^n$,

$$x \xrightarrow[\mathrm{a}]{f} y \iff \forall i \in \Delta(x, y), y_i = f_i(x) \qquad [2]$$

where $\Delta(x, y)$ is the list of components which state differs between x and y, i.e., $\Delta(x, y) = \{i \in \{1, \ldots, n\} \mid x_i \neq y_i\}$.

A configuration $y \in \mathbb{B}^n$ is *reachable* from $x \in \mathbb{B}^n$ if either $x = y$, or there exists a sequence of transitions from x to y:

$$\rho^f_{\mathrm{a}}(x) = \{y \in \mathbb{B}^n \mid x \xrightarrow[\mathrm{a}]{f}{}^{*} y\} \ . \qquad [3]$$

Notice that if $y \notin \rho^f_{\mathrm{a}}(x)$, then it is impossible to evolve from x to y according to any of the semantics defined above, including the synchronous and (fully) asynchronous ones.

Fig. 2(c) shows all possible asynchronous evolutions of the example BN from the configuration where all the components are inactive, i.e. $\rho^f_{\mathrm{a}}(000) = \{000, 110, 010, 011, 100\}$.

Reachability is a fundamental property to assess the compatibility of BN models with time series data: if none of the configurations matching an observation at a given time is reachable from any configuration matching an experimental observation at an earlier time, the BN cannot predict the observed behaviors.

Another prominent dynamical property studied with BNs, strongly linked to reachability, are attractors. Attractors represent the long-term behaviors of the model and are often used to represent cell phenotypes. Formally, an attractor is a smallest non-empty set of configurations from which it is impossible to escape: $A \subseteq \mathbb{B}^n$ is an attractor if and only each of its configuration $z \in A$ verifies $\rho^f_{\mathrm{a}}(z) = A$. An attractor is said to be a fixed point whenever it is a single configuration $z \in \mathbb{B}^n$ (whenever $f(z) = z$ with the asynchronous semantics), and *complex* if it is an ensemble of configurations, such as *cyclic* attractors, modeling potential sustained oscillations.

Refinements of Boolean Networks

Boolean Networks impose a drastic coarse-graining on component activity. Several modeling frameworks introduced a finer granularity in logical models (17). Examples include Multivalued Networks (MN) (18), where components can take more than two *logical* values (0, 1, 2, . . . , m), fuzzy logic (19), which extends logical models with continuous domains, stochastic extensions of fully asynchronous Boolean Networks (20), and

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.22.998377. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

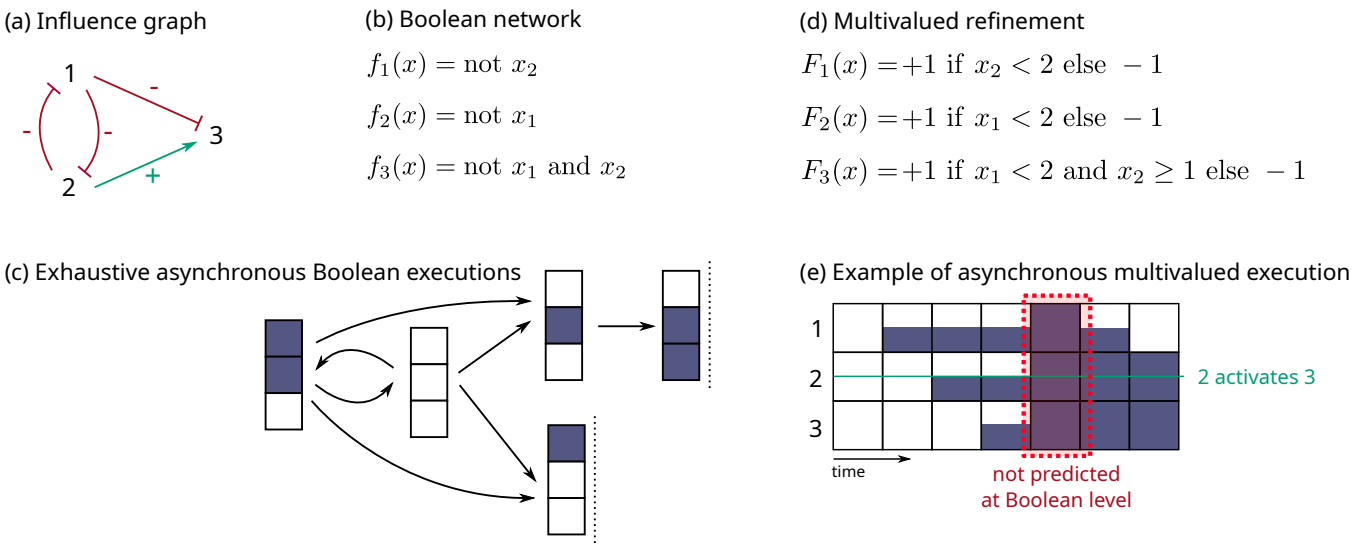

Fig. 2. Example of qualitative models for the interactions between three components. (a) Influence graph denoting the activation and inhibition relationships. (b) Example of a compatible BN, which defines the activation conditions of each component. (c) Exhaustive list of transitions obtained from the initial configuration where all three components are inactive. (d) Example of a multivalued network refining the BN (b) with components able to exist under three states (0, 1, 2). (e) Example of asynchronous execution of the multivalued network from the configuration 000. Half and fully blue squares represent the states 1 and 2.

Ordinary Differential Equations (ODEs) (21, 22), where values of components are non-negative reals and vary along continuous time. Their specifications require, however, much more information about the biological system, such as thresholds of interactions for MNs and precise kinetics for ODEs. These parameters are often unknown, and their automatic inference would require a significant amount of data collected in similar experimental settings.

One could use any of these frameworks to model the same biological system at different abstraction levels. Which raises the question of the relationship between models from different frameworks: is an MN model F a refinement of a BN model f? In other words, does F specify a system with more quantitative information than f, but follows the same (Boolean) logic for the interactions.

We consider here a simple mathematical criterion for refinements: the value of a component can decrease (resp. increase) only if the component can be set to 0 (resp. 1) in the BN with a possible binarization of the state. A formal definition will be given in the next section.

Incompleteness of (a)synchronous Boolean networks. We can define a multivalued network F of dimension n by a discrete function, which maps, for each component, states to the tendency of value change (decrease, steady, increase). To ease notations, and without loss of generality, we assume that all the components can take an integer value between 0 and the same fixed m:

$$F : \mathbb{M}^n \to \{-1, 0, 1\}^n \qquad [4]$$

where $\mathbb{M} = \{0, 1, \ldots, m\}$. The successors of a configuration $x \in \mathbb{M}^n$ are then computed by adding the value of $F(x)$ to (a subset of) components, provided they stay non-negative and do not exceed their maximum value m.

In Fig. 2 we present a simple example of BN for which asynchronous executions miss possible behaviors of the network when considering a multivalued refinement of it. The MN in Fig. 2(d) is a refinement of the BN in Fig. 2(b). In addition to the higher granularity for the activity levels of all three components, it brings additional information on the activation of component 3. An intermediate value of 2 is sufficient to activate 3 provided that the value of its inhibitor 1 is not high. One of its asynchronous execution shown in Fig. 2(e) predicts that the three components can get activated simultaneously, which was never predicted by any of the asynchronous executions of the BN. Assuming the validation of the model were subject to the reachability of a configuration with all the three components active from a configuration with all the components inactive, this BN model would be deemed insufficient for achieving the observed behavior, with an erroneous conclusion that its underlying influence graph is wrong.

A new execution paradigm for Boolean networks

The critical reason usual BN interpretations miss behaviors is that the binary coarse-graining coupled with the instantaneous state changes preempt interactions occurring *during* the course of (de)activations. In the counterexample of Fig. 2, Boolean interpretations exclude the activation of component 3 during the activation of components 1 and 2, whereas, in a possible refinement, 3 can indeed increase before 2 reaches its fully active state and before 1 is sufficiently expressed to inhibit it.

The Most Permissive semantics. We devised a new dynamical interpretation of BNs, called Most Permissive semantics, in which we consider that a component can exist in 4 states: inactive (0), increasing (↗), decreasing (↘), or active (1). While a component is in an dynamic state (increasing or decreasing), it can be read non-deterministically as either 0 or 1. These ambiguous states account for the absence of information on actual influence thresholds: a component in a dynamic state can be above the influence threshold for one component while being below the influence threshold for another one.

Fig. 3 illustrates the changes of component states possible

DRAFT

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.22.998377. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

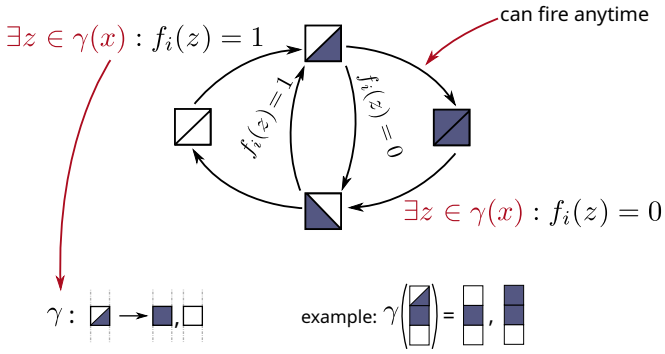

Fig. 3. Conditions for the changes of component states $i \in \{1, \ldots, n\}$ in Most Permissive Boolean Networks from a most-permissive configuration $x \in \mathbb{P}^n$ with $\mathbb{P} = \{0, \nearrow, \searrow, 1\}$. The increasing state $\nearrow$ is represented by a top-left white and bottom-right blue square, the decreasing state $\searrow$ by a bottom-left blue and top-right white square. The function γ gives the admissible Boolean interpretations of $x \in \mathbb{P}^n$: $\gamma(x) = \{z \in \mathbb{B}^n \mid \forall i \in \{1, \ldots, n\}, x_i \in \mathbb{B} \Rightarrow z_i = x_i\}$, *i.e.*, all the components in Boolean states are fixed, and the others are free.

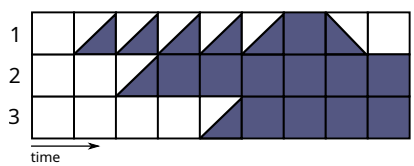

Fig. 4. One of the possible executions using the Most Permissive semantics on the Boolean network in Fig. 2(b) starting from the configuration where all genes are inactive. Note that it correctly recovers the (transient) reachability of the configuration where the three components are active.

with the Most Permissive semantics. A component i can change to the increasing (resp. decreasing) state whenever it can interpret the value of its regulators in a way which makes its logical function f_i true (resp. false) – if one of its regulators is in a dynamic state, both Boolean interpretations can be considered. Once in increasing (resp. decreasing) state, it can reach 1 (resp. 0) at any time. Each component evolves independently of all others. The complete formal definition is given in SI.2.

Fig. 4 shows an example of execution using the Most Permissive semantics on the BN of Fig. 2. Contrary to the (a)synchronous interpretations, the Most Permissive semantics correctly captures the possible (transient) reachability of the configuration where the three genes are active. While component 1 is "increasing" and component 2 is active, gene 3 can indeed change to "increasing", thus leading to the activation of all three components. This configuration is not in an attractor, and both single-point attractors identified in Fig. 2(c) are reachable via different Most Permissive executions. We provide the application of MPBNs to the BN presented in Fig. 1 in in Fig. S1.

Formal guarantees for model refinements. Using the simple examples in Fig. 1 and Fig. 2, we have shown that BN refinements can introduce behaviors that cannot be captured with classical semantics.

Most Permissive Boolean Networks bring the formal guarantee of being able to reproduce *all* the behaviors achievable in *any* refinements, being a multivalued network or an ODE system (Theorem 1 and Corollary 1 in SI.2). In other words, if the Most Permissive semantics concludes that it is impossible to observe a given state change for some components, then no qualitative or quantitative model verifying the refinement criteria can predict these state changes.

The refinement criterion relies on a *binarization* of the multivalued configuration. An appropriate binarization necessarily quantifies 0 as Boolean 0 and m as 1, and is free for the other intermediate values. Let us denote by $\beta(x)$ the set of possible binarization of configuration $x \in \mathbb{M}^n$:

$$\beta(x) = \{x' \in \mathbb{B}^n \mid \forall i \in \{1, \ldots, n\}, x_i = 0 \Rightarrow x'_i = 0 \text{ and } x_i = m \Rightarrow x'_i = 1\} \ . \quad [5]$$

For example with $m = 2$, $\beta(012) = \{001, 011\}$.

Then, we say a MN F is a refinement of a BN f of the same dimension n if and only if for every configuration $x \in \mathbb{M}^n$, and for every component $i \in \{1, \ldots, n\}$, $F_i(x) < 0$ there exists $x' \in \beta(x)$ such that $f_i(x') = 0$, and $F_i(x) > 0$ implies there exists $x' \in \beta(x)$ such that $f_i(x') = 1$.

This characterization of BN refinement to MN can be directly extended to ODEs. Indeed, ODEs specify the (real) derivative of the (positive real) value of each component:

$$\mathcal{F} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n \ . \quad [6]$$

Only the binarization β should be adapted in Eq. (5) to reflect that there is no (a priori) upper bounded value m for components.

The completeness property states the following. Consider a multivalued refinement F of a BN f with which there exists an asynchronous trajectory from a multivalued configuration x to y. Let us write $\hat{x}$ any most-permissive configuration *compatible* with x: if $x_i = 0$, then $\hat{x}_i = 0$, if x_i is the maximum value of i, then $\hat{x}_i = 1$, and in the other cases $\hat{x}_i$ can be either $\nearrow$ or $\searrow$. Then, there exists a most-permissive trajectory leading to any of these $\hat{x}$ to a most-permissive configuration $\hat{y}$ compatible with y and which is consistent with the the changes between x and y: $\hat{y}_i = \nearrow$ if $y_i > x_i$ and $y_i < m$, $\hat{y}_i = \searrow$ if $y_i < x_i$ and $y_i > 0$, and $\hat{y}_i = \hat{x}_i$ if $y_i = x_i$. As the proof relies solely on the sign of the derivative of the refinement of f, the property extends to ODE refinements, which can be seen as multivalued networks with m to infinity.

Allowing any state change without restriction would also provide the above guarantee. It appears that if there is a most-permissive trajectory between two binary configurations, then there is a multilevel refinement of the BN showing an asynchronous trajectory between matching multilevel configurations (Theorem 2 in SI.2). Therefore, the completeness property can be achieved only by predicting *at least* the behaviors of MPBNs. In other words, the Most Permissive semantics is the tightest Boolean abstraction of multivalued refinements regarding reachability properties.

Simpler computational complexity. Most computational analyzes of BNs focus on two elementary dynamical properties: the reachability, which is the existence of a trajectory between two given configurations, and the existence of attractors. Here, we study these properties in term of algorithmic complexity classes. These theoretical results have very concrete implications for the analysis of MPBNs, making the approach scalable to genome-scale networks.

We first recall the bases of computational complexity classes (23): the P class contains algorithms running in time polynomial with the size of its inputs; the NP class contains

DRAFT

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.22.998377. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

algorithms running in polynomial time with non-deterministic choices; the PSPACE class contains algorithms running in polynomial *space*. We know that $\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE}$, where "$\subseteq$" can be understood as "simpler". A problem is complete for a given complexity class if it belongs to and is among the hardest problems of this class. The famous "SAT" problem of determining if a formula expressed in propositional logic (essentially Boolean variables and logic connectors) has a satisfying solution is NP-complete. We do not know if NP = PSPACE, but in practice, NP-complete problems are much more tractable than PSPACE-complete ones by several orders of magnitude. Hereafter, we also refer to the coNP class, which consists of problems for which finding a counter-example belongs to NP, and to the P^{NP} and $\mathrm{coNP}^{\mathrm{coNP}}$ classes, ordered as follows: $\mathrm{coNP} \subseteq \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{coNP}^{\mathrm{coNP}} \subseteq \mathrm{PSPACE}$.

With asynchronous BNs, it is challenging to determine if a trajectory exists between two configurations since, in the worst case, it requires exploring all the possible configurations. With MPBNs, this problem is much simpler thanks to an intriguing property: if there exists a trajectory between two configurations, then there is such a trajectory visiting at most $3n$ configurations. Intuitively, this shortcut corresponds to a particular sequence of state changes: in a first phase, only transitions changing a state from 0 or 1 to $\nearrow$ or $\searrow$ take place; In a second phase, only transitions changing states within $\nearrow$ and $\searrow$; in a final phase, only transitions changing states from $\nearrow$ or $\searrow$ to 1 or 0. Each phase comprises at most n transitions, one for each component.

Moreover, finding this shortcut requires exploring at most a quadratic number of transitions in the general case, and only $3n$ whenever the target configuration is in an attractor. The exploration consists of performing as many transitions as possible of the first phase, putting the largest possible number of components in a dynamic state. For each component whose state does not change between the starting and target configuration, it is then necessary to switch the dynamic state back (second phase). If this is not possible, then the exploration is repeated from the beginning while preventing this specific component from changing to a dynamic state (as it would still be impossible to go back to the initial binary state, and the target configuration would not be part of an attractor). Overall, the exploration is thus repeated at most n times. Finally, all the transitions of the third phase are applied, which should lead to the target configuration if and only if it is reachable.

On the other hand, determining the possibility of a most-permissive transition is NP-complete in the general case: indeed, the condition "$\exists z \in \gamma(x) : f_i(z) = 1$" in Fig. 3 is the SAT problem. For biological networks, it is usual to assume that components cannot have both positive (activator) and negative (inhibitor) direct influences. The resulting BNs are called *locally monotonic*: each local function f_i is monotonic for every component it depends on: increasing the number of activators (resp. inhibitors) of i in state 1 can only increase (resp. decrease) the value of f_i. Thus, determining the existence of a Boolean interpretation z of a most-permissive configuration x so that $f_i(z) = 1$ comes down to considering activators in dynamic state as 1 and inhibitors in dynamic state as 0, and conversely for $f_i(z) = 0$. Therefore, determining the possibility of a most-permissive transition can be done in linear time with locally-monotonic BNs.

The reachability problem in MPBNs can thus be solved in polynomial time whenever f is locally monotonic (Theorem 3 in SI.2), a considerable drop in complexity compared to synchronous or asynchronous BNs where the problem is PSPACE-complete. With non-locally monotonic BNs, the reachability problem is in P^{NP}.

While the attractors of asynchronous BNs can be complex objects, the attractors of MPBNs are particular mathematical objects called *minimal trap spaces*. A trap space is a hypercube which is closed by f: for any vertex x, $f(x)$ is also a vertex. A trap space is minimal whenever it does not include a different trap space. Attractors in MPBNs have this regular structure because whenever two configurations lying on any diagonal of an hypercube are reachable from each other, they can reach the adjacent configurations as well.

Determining if a configuration $x \in \mathbb{B}^n$ belongs to an attractor of f is a key problem to identify attractors of a BNs. It is again a PSPACE-complete problem for synchronous and asynchronous BNs. In the case of MPBNs, it boils down to verifying if the trap space containing x is minimal, which is at most of complexity coNP for locally monotonic BNs, and at most $\mathrm{coNP}^{\mathrm{coNP}}$ for non-locally monotonic BNs (Theorem 4 in SI.2). The computation of minimal trap spaces of a BN can be performed efficiently with SAT solvers and related logic programming frameworks (24). On a regular 3.3GHz processor, our implementation of MPBNs can compute reachable attractors of randomly generated scale-free networks (25) with 1,000 components in a fraction of a second, less than 2 seconds with 10,000 components, and less than 50 seconds with 100,000 components (SI.3.C, Fig. S4).

Validation of MPBNs on actual biological models. An essential feature of logical models is their ability to conclude on the *absence* of certain behaviors. For instance, differentiation processes are modeled using separate attractors representing the final phenotypes and trajectories where configurations are committed to reaching a particular attractor with no possibility to rejoin other differentiation branches. A model allowing any configuration to reach any attractor would indeed be useless without quantitative aspects. We will show that, although enabling more behaviors than (a)synchronous BNs, MPBNs are still constraining and able to capture differentiation and cell fate decisions.

As we have said above, attractors of MPBNs correspond to the minimal trap spaces of the Boolean function. Prior work has shown that these trap spaces match well with the complex attractors of fully asynchronous BNs in many real-world models of biological networks (24). To further assess how MPBNs perform in practice, we reproduced studies on logical models of differentiation using MPBNs instead of fully asynchronous BNs (SI.3).

In the case of a tumor invasion model (8), MPBNs correctly predicted the loss of reachability of apoptotic attractors upon the mutations of p53 and NICD presented in the study (Fig. S2). In the case of T-cell differentiation (9), MPBNs recovered the same reprogramming graph between T-cell types (Fig. S3). After booleanization (26), MPBNs efficiently handled the original large multivalued model of 100 species, whereas the original study had to perform approximations through model reduction. Therefore, the Most Permissive interpretation of BNs is still stringent enough to capture processes that control reachable attractors.

In conclusion, MPBNs are formally guaranteed never to

DRAFT

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.22.998377. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

ignore behaviors hidden by artifacts of usual Boolean modeling while still being specific enough to predict differentiation processes, and doing so at a much lower computational cost.

Discussion

The choice of the dynamical interpretation of BNs has drastic effects on their predictions. Whereas the (fully) asynchronous BN interpretation is often advised for practical applications, it overlooks behaviors emerging from different timescales for the interactions, leading to biases when selecting plausible models. Such misses are due to artifacts of configurations updates. On the contrary, MPBNs offer a framework for reasoning on the qualitative dynamics without making any strong *a priori* hypothesis about the timescale and thresholds of interactions, and without additional parameter.

The state-space explosion triggered by the usual interpretations of BNs is another significant bottleneck for their application in systems biology (3, 27). MPBNs offer drastic gains in computational complexity when analyzing possible trajectories and attractors, both elementary and essential properties, underpining the potential of a model. In practice, the verification of these properties with asynchronous BNs is typically limited to networks with 50 to 100 nodes. On the contrary, deciding the reachability and attractor properties in MPBNs relies on scalable algorithms and does not suffer from the state-space explosion. For the case of locally-monotonic BNs, which is a classical hypothesis for biological networks, the complexity allows addressing very large scale networks, as illustrated in SI.3, with experiments on BNs with up to 100,000 components. Our software tool `mpbn` is available at https://github.com/pauleve/mpbn and integrated in the CoLoMoTo notebook environment (28).

The prediction of attractors reachable from specific initial conditions, and possibly under various mutant conditions, is at the core of many studies using logical models. While MPBNs can identify the complete set of reachable attractors several orders of magnitude faster than asynchronous BNs, the quantification of the propensities of each attractor, e.g., performed by sampling the trajectories (20, 29), is yet to be explored. In addition to the validation of and the control of predictions from genome-scale models, the complexity breakthrough brought by MPBNs together with their ability to overcome artifacts of Boolean modeling paves the way towards the inference and learning of large-scale logical models from experimental data.

ACKNOWLEDGMENTS. This research was funded by the French Agence Nationale pour la Recherche (ANR) in the context of ANR-FNR project "AlgoReCell" (ANR-16-CE12-0034)

Authors' contributions. LP, TC, SH designed the research; LP defined MPBNs and demonstrated Theorems 1,3,4, implemented code, performed experiments, and initially drafted the manuscript; JK demonstrated Theorem 2; LP, JK, TC, SH wrote the manuscript.

References

1. S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969. doi:10.1016/0022-5193(69)90015-0.
2. R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563 – 585, 1973. doi:10.1016/0022-5193(73)90247-6.
3. N. Le Novère. Quantitative and logic modelling of molecular and gene networks. *Nature reviews. Genetics*, 16:146–158, 2015. doi:10.1038/nrg3885.
4. S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein. Random Boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences*, 100(25):14796–14799, 2003. doi:10.1073/pnas.2036429100.
5. R. Zhang, M. V. Shah, J. Yang, S. B. Nyland, X. Liu, J. K. Yun, R. Albert, and T. P. Loughran. Network model of survival signaling in large granular lymphocyte leukemia. *Proceedings of the National Academy of Sciences*, 105(42):16308–16313, 2008. doi:10.1073/pnas.0806447105.
6. Z. Mai and H. Liu. Boolean network-based analysis of the apoptosis network: Irreversible apoptosis and stable surviving. *Journal of Theoretical Biology*, 259(4):760 – 769, 2009. doi:https://doi.org/10.1016/j.jtbi.2009.04.024.
7. P. Martínez-Sosa and L. Mendoza. The regulatory network that controls the differentiation of t lymphocytes. *Biosystems*, 113(2):96 – 103, 2013. doi:https://doi.org/10.1016/j.biosystems.2013.05.007.
8. D. P. A. Cohen, L. Martignetti, S. Robine, E. Barillot, A. Zinovyev, and L. Calzone. Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Comput Biol*, 11(11):e1004571, 2015. doi:10.1371/journal.pcbi.1004571.
9. W. Abou-Jaoudé, P. T. Monteiro, A. Naldi, M. Grandclaudon, V. Soumelis, C. Chaouiya, and D. Thieffry. Model checking to assess T-helper cell plasticity. *Frontiers in Bioengineering and Biotechnology*, 2, 2015. doi:10.3389/fbioe.2014.00086.
10. P. Traynard, A. Fauré, F. Fages, and D. Thieffry. Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation. *Bioinformatics*, 32(17):i772–i780, 2016. doi:10.1093/bioinformatics/btw457.
11. S. Collombet, C. van Oevelen, J. L. Sardina Ortega, W. Abou-Jaoudé, B. Di Stefano, M. Thomas-Chollier, T. Graf, and D. Thieffry. Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proceedings of the National Academy of Sciences*, 114(23):5792–5799, 2017. doi:10.1073/pnas.1610622114.
12. R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. i. structural conditions of multistationarity and other nontrivial behavior. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11(1):170, 2001. doi:10.1063/1.1350439.
13. C. Soulé. Graphic requirements for multistationarity. *Complexus*, 1(3):123–133, 2003. doi:10.1159/000076100.
14. L. Paulevé and A. Richard. Static analysis of Boolean networks based on interaction graphs: a survey. *Electronic Notes in Theoretical Computer Science*, 284:93 – 104, 2011. doi:10.1016/j.entcs.2012.05.017.
15. S. Soliman. A stronger necessary condition for the multistationarity of chemical reaction networks. *Bulletin of Mathematical Biology*, 75(11):2289–2303, 2013. doi:10.1007/s11538-013-9893-7.
16. S. Mangan and U. Alon. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985, 2003. doi:10.1073/pnas.2133841100.
17. M. K. Morris, J. Saez-Rodriguez, P. K. Sorger, and D. A. Lauffenburger. Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15):3216–3224, 2010. doi:10.1021/bi902202q.
18. R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, Boca Raton , Florida, USA, 1990.
19. B. B. Aldridge, J. Saez-Rodriguez, J. L. Muhlich, P. K. Sorger, and D. A. Lauffenburger. Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/insulin-induced signaling. *PLoS Computational Biology*, 5(4):e1000340, 2009. doi:10.1371/journal.pcbi.1000340.
20. G. Stoll, E. Viara, E. Barillot, and L. Calzone. Continuous time boolean modeling for biological signaling: application of gillespie algorithm. *BMC Systems Biology*, 6(1):116, 2012. doi:10.1186/1752-0509-6-116.
21. L. Glass and S. Kauffman. Logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology*, 39(1):103–129, 1973. doi:10.1016/0022-5193(73)90208-7.
22. H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9:67–103, 2002. doi:10.1089/10665270252833208.
23. C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1995.
24. H. Klarner, A. Bockmayr, and H. Siebert. Computing maximal and minimal trap spaces of Boolean networks. *Natural Computing*, 14(4):535–544, 2015. doi:10.1007/s11047-015-9520-7.
25. R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002. doi:10.1103/revmodphys.74.47.
26. G. Didier, E. Remy, and C. Chaouiya. Mapping multivalued onto Boolean dynamics. *Journal of Theoretical Biology*, 270(1):177 – 184, 2011. doi:10.1016/j.jtbi.2010.09.017.
27. S. Bornholdt. Boolean network models of cellular regulation: prospects and limitations. *Journal of The Royal Society Interface*, 5(suppl_1), 2008. doi:10.1098/rsif.2008.0132.focus.
28. A. Naldi, C. Hernandez, N. Levy, G. Stoll, P. T. Monteiro, C. Chaouiya, T. Helikar, A. Zinovyev, L. Calzone, S. Cohen-Boulakia, D. Thieffry, and L. Paulevé. The CoLoMoTo Interactive Notebook: Accessible and Reproducible Computational Analyses for Qualitative Biological Networks. *Frontiers in Physiology*, 9:680, 2018. doi:10.3389/fphys.2018.00680.
29. N. D. Mendes, R. Henriques, E. Remy, J. Carneiro, P. T. Monteiro, and C. Chaouiya. Estimating attractor reachability in asynchronous logical models. *Frontiers in Physiology*, 9, 2018. doi:10.3389/fphys.2018.01161.

DRAFT