Reducing Boolean Networks with Backward Boolean Equivalence*

Georgios Argyris[1][0000−0002−3203−0410], Alberto Lluch Lafuente[1][0000−0001−7405−0818], Mirco Tribastone[2][0000−0002−6018−5989], Max Tschaikowski[3][0000−0002−6186−8669], and Andrea Vandin[4,1][0000−0002−2606−7241]

[1] DTU Technical University of Denmark, Kongens Lyngby, Denmark
[2] IMT School for Advanced Studies Lucca, Italy
[3] University of Aalborg, Denmark
[4] Sant'Anna School for Advanced Studies, Pisa, Italy

Abstract. Boolean Networks (BNs) are established models to qualitatively describe biological systems. The analysis of BNs might be infeasible for medium to large BNs due to the state-space explosion problem. We propose a novel reduction technique called *Backward Boolean Equivalence* (BBE), which preserves some properties of interest of BNs. In particular, reduced BNs provide a compact representation by grouping variables that, if initialized equally, are always updated equally. The resulting reduced state space is a subset of the original one, restricted to identical initialization of grouped variables. The corresponding trajectories of the original BN can be exactly restored. We show the effectiveness of BBE by performing a large-scale validation on the whole GINsim BN repository. In selected cases, we show how our method enables analyses that would be otherwise intractable. Our method complements, and can be combined with, other reduction methods found in the literature.

Keywords: Boolean Network · State Transition Graph · Attractor Analysis · Exact Reduction · GinSim Repository

1 Introduction

Boolean Networks (BNs) are an established method to model biological systems [28]. A BN consists of Boolean variables (also called nodes) which represent the activation status of the components in the model. The variables are commonly depicted as nodes in a network with directed links which represent influences between them. However, a full descriptive mathematical model underlying a BN consists of a set of Boolean functions, the *update functions*, that govern the Boolean values of the variables. Two BNs are displayed on top of Fig. 1. The BN on the left has three variables x_1, x_2, and x_3, and the BN on the right has two variables $x_{1,2}$ and x_3. The dynamics (the state space) of a BN

* Partially supported by the DFF project REDUCTO 9040-00224B, the Poul Due Jensen Foundation grant 883901, and the PRIN project SEDUCE 2017TWRCNB.

is encoded into a *state transition graph* (STG). The bottom part of Fig. 1 displays the STGs of the corresponding BNs. The boxes of the STG represent the BN *states*, i.e. vectors with one Boolean value per BN variable. A directed edge among two STG states represents the evolution of the system from the source state to the target one. The target state is obtained by synchronously applying all the update functions to the activation values of the source state. There exist BN variants with other update schema, e.g. asynchronous non-deterministic [47] or probabilistic [43]. Here we focus on the synchronous case. BNs where variables are *multivalued*, i.e. can take more than two values to express different levels of activation [46], are supported via the use of *booleanization* techniques [18], at the cost, however, of increasing the number of variables.

$$\begin{array}{l} x_1(t+1) = \neg x_3(t) \lor x_1(t) \\ x_2(t+1) = x_1(t) \lor x_2(t) \lor \neg x_3(t) \\ x_3(t+1) = x_2(t) \land \neg x_3(t) \end{array} \xRightarrow[x_1, x_2 \,:\, BBE]{} \begin{array}{r} x_{1,2}(t+1) = \neg x_3(t) \lor x_{1,2}(t) \\ x_3(t+1) = x_{1,2}(t) \land \neg x_3(t) \end{array}$$

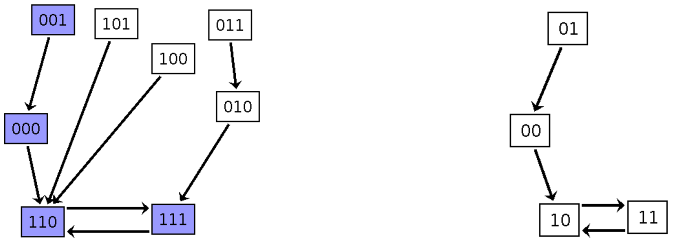


Fig. 1: A BN (top-left), its STG (bottom-left), the BBE-reduced BN (top-right) and its (reduced) STG (bottom-right).

BNs suffer from the state space explosion problem: there are exponentially many STG states with respect to the number of BN variables. This hampers BN analysis in practice, calling for reduction techniques for BNs. There exist manual or semi-automated ones based on domain knowledge. Such empirical reductions have several drawbacks: being semi-automated, they are error-prone, and do not scale. Popular examples are those based on the idea of *variable absorption*, proposed originally in [34, 48, 41]. The main idea is that certain BN variables can get *absorbed* by the update functions of their target variables by replacing all occurrences of the absorbed variables with their update functions. Other methods automatically remove *leaf* variables (variables with 0 outgoing links) or *frozen* variables (variables that stabilize after some iterations independently of the initial conditions) [39, 4]. Several techniques [23, 2] focus on reducing the STGs rather than the BN generating them. This requires to construct the original STG, thus still incurring the state space explosion problem.

Our research contributes a novel mathematically grounded method to automatically minimize BNs while exactly preserving behaviors of interest. We present Backward Boolean Equivalence (BBE), which collapses *backward Boolean equivalent* variables. The main intuition is that two BN variables are BBE-equivalent if they maintain equal value in any state reachable from a state wherein they have the same value. In the STG in Fig. 1 (left), we note that for all states where x_1 and x_2 have same value (purple boxes), the update functions do not distinguish them. Notably, BBE is that it can be checked directly on the BN, without requiring to generate the STG. Indeed, as depicted in the middle of Fig. 1 , x_1 and x_2 can be shown to be BBE-equivalent by inspecting their update functions: If x_1, x_2 have the same value in a state, i.e. $x_1(t) = x_2(t)$, then their update functions will not differentiate them since $x_2(t+1) = x_1(t) \vee x_2(t) \vee \neg x_3(t) = x_1(t) \vee x_1(t) \vee \neg x_3(t) = x_1(t) \vee \neg x_3(t) = x_1(t+1)$. We also present an iterative partition refinement algorithm [36] that computes the largest BBE of a BN. Furthermore, given a BBE, we obtain a *BBE-reduced* BN by collapsing all BBE-equivalent variables into one in the reduced BN. In Fig. 1, we collapsed x_1, x_2 into $x_{1,2}$. The reduced BN faithfully preserves part of the dynamics of the original BN: it exactly preserves all states and paths of the original STG where BBE-equivalent variables have same activation status. Fig. 1 (right) shows the obtained BBE-reduced BN and its STG. We can see that the purple states of the original STG are preserved in the one of the reduced BN.

We implemented BBE in ERODE [10], a freely available tool for reducing biological systems. We built a toolchain that combines ERODE with several tools for the analysis, visualization and reduction of BNs, allowing us to apply BBE to all BNs from the GINsim repository (http://ginsim.org/models_repository). BBE led to reduction in 61 out of 85 considered models (70%), facilitating STG generation. For two models, we could obtain the STG of the reduced BN while it is not possible to generate the original STG due to its size. We further demonstrate the effectiveness of BBE in three case studies, focusing on their *asymptotic dynamics* by means of *attractors analysis*. Using BBE, we can identify the attractors of large BNs which would be otherwise intractable.

The article is organized as follows: Section 2 provides the basic definitions and the running example based on which we will explain the key concepts. In Section 3 , we introduce BBE, present the algorithm for the automatic computation of maximal BBEs, and formalize how the STGs of the original and the reduced BN are related. In Section 4, we apply BBE to BNs from the literature. In Section 5 we discuss related works, while Section 6 concludes the paper.

2 Preliminaries

BNs can be represented visually using some graphical representation which, however, might not contain all the information about their dynamics [29]. An example is that of signed interaction (or regulatory) graphs adopted by the tool GinSim [31]. These representations are often paired with a more precise description containing either truth tables [39] or algebraic update functions [45]. In this paper we focus on such precise representation, and in particular on the latter. However, in order to better guide the reader in the case studies, wherein we manipulate BNs with a very large number of components, we also introduce signed interaction graphs.

$$\begin{aligned}
x_{Fgf8}(t+1) &= x_{Fgf8}(t) \land \neg x_{Emx2}(t) \land x_{Sp8}(t)\\
x_{Pax6}(t+1) &= \neg x_{Emx2}(t) \land x_{Sp8}(t) \land \neg x_{Coup\_tfi}(t)\\
x_{Emx2}(t+1) &= \neg x_{Fgf8}(t) \land \neg x_{Pax6}(t) \land \neg x_{Sp8}(t) \land x_{Coup\_tfi}(t)\\
x_{Sp8}(t+1) &= x_{Fgf8}(t) \land \neg x_{Emx2}(t)\\
x_{Coup\_tfi}(t+1) &= \neg x_{Fgf8}(t) \land \neg x_{Sp8}(t)
\end{aligned}$$

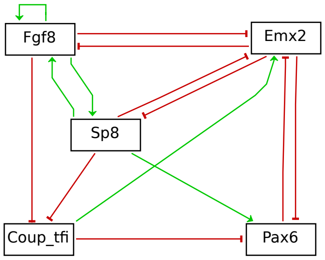


Fig. 2: (Left) the BN of cortical area development from [25]; (Right) its signed interaction graph.

We explain the concepts of current and next sections using the simple BN of Fig. 2 (left) taken from [25]. The model refers to the development of the outer part of the brain: the cerebral cortex. This part of the brain contains different areas with specialised functions. The BN is composed of five variables which represent the gradients that take part in its development: the morphogen *Fgf8* and four transcription factors, i.e., *Emx2, Pax6, Coup_tfi, Sp8*. During development, these genes are expressed in different concentrations across the surface of the cortex forming the different areas.

Fig. 2 (right) displays the signed interaction graph that corresponds to the BN. The green arrows correspond to *activations* whereas the red arrows correspond to *inhibitions*. For example, the green arrow from *Sp8* to *Pax6* denotes that the former promotes the latter because variable x_{Sp8} appears (without negation) in the update function of x_{Pax6}, whereas the red arrow from *Pax6* to *Emx2* denotes that the former inhibits the latter because the negation of x_{Pax6} appears in the update function of x_{Emx2}.

We now give the formal definition of a BN:

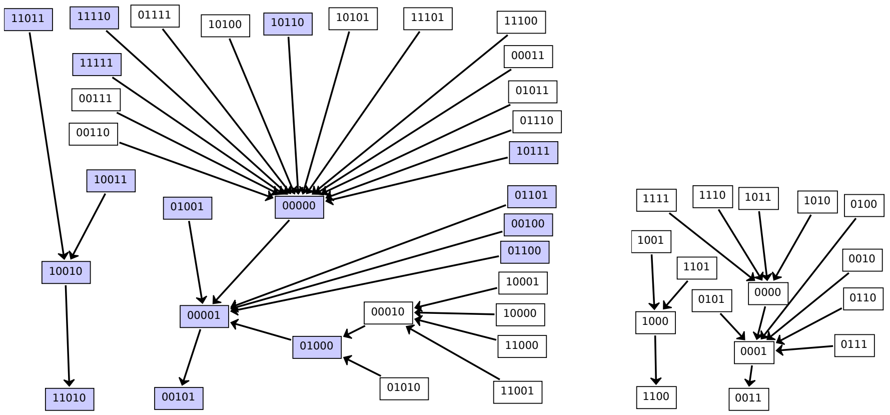
Definition 1. *A BN is a pair (X, F) where $X = \{x_1, ..., x_n\}$ is a set of variables and $F = \{f_{x_1}, ..., f_{x_n}\}$ is a set of update functions, with $f_{x_i} : \mathbb{B}^n \to \mathbb{B}$ being the update function of variable x_i.*

A BN is often denoted as $X(t+1) = F(X, t)$, or just $X = F(X)$. In Fig. 2 we have $X = \{x_{Fgf8}, x_{Pax6}, x_{Emx2}, x_{Sp8}, x_{Coup\_tfi}\}$.

The *state* of a BN is an evaluation of the variables, denoted with the vector of values $\mathbf{s} = (s_{x_1}, \ldots, s_{x_n}) \in \mathbb{B}^n$. The variable x_i has the value s_{x_i}. When the update functions are applied synchronously, we have synchronous transitions between states, i.e. for $\mathbf{s}, \mathbf{t} \in \mathbb{B}^n$ we have $\mathbf{s} \to \mathbf{t}$ if $\mathbf{t} = F(\mathbf{s}) = (f_{x_1}(\mathbf{s}), \ldots, f_{x_n}(\mathbf{s}))$.

Suppose that the activation status of the variables x_{Fgf8}, x_{Emx2}, x_{Pax6}, x_{Sp8}, x_{Coup_tfi} is given by the state $\mathbf{s} = (1, 0, 1, 1, 1)$. After applying the update functions, we have $\mathbf{t} = F(\mathbf{s}) = (0, 0, 0, 0, 0)$.

The state space of a BN, called *State Transition Graph (STG)*, is the set of all possible states and state transitions.



(a) STG of BN in Fig. 2 (b) STG of BN in Fig. 4

Fig. 3: The STGs of the BN of Fig. 2 and of its BBE-reduction in. Fig. 4. We use GINsim's visual representation, where self-loops are implicit in nodes without outgoing edges.

Definition 2. *Let $B = (X, F)$ be a BN. We define the state transition graph of B, denoted with $STG(B)$, as a pair (S, T) with $S \subseteq \mathbb{B}^n$ being a set of vertices labelled with the states of B, and $T = \{\boldsymbol{s} \to \boldsymbol{t} \mid \boldsymbol{s} \in S, \boldsymbol{t} = F(\boldsymbol{s})\}$ a set of directed edges representing the transitions between states of B.*

We often use the notation $\mathbf{s} \to^{+} \mathbf{t}$ for the transitive closure of the transition relation. The cardinality of the set of states is 2^n, which illustrates the state space explosion: we have exponentially many states on BN variables. Fig. 3(a) displays the STG of the BN in Fig. 2.

Several BN properties are identified in STGs, e.g. attractors, basins of attraction, and transient trajectories [42]. Attractors are sets of states towards which a system tends to evolve and remain [27]. They are often associated with the interpretation of the underlying system; for example, Kauffman equated attractors with different cell types [20]. Hence, the main reduction methods that have been developed in the literature so far concentrate on how they affect the asymptotic dynamics i.e. the number of attractors and the distribution of their lengths. We define an attractor as follows:

Definition 3. ***(Attractor)*** *Let $B = (X, F)$ be a BN with $STG(B) = (S, T)$. We say that a set of states $A \subseteq S$ is an* attractor *iff*

1. $\forall \boldsymbol{s}, \boldsymbol{s}' \in A$, $\boldsymbol{s} \to^{+} \boldsymbol{s}'$, and
2. $\forall \boldsymbol{s} \in A, \forall \boldsymbol{s}' \in S$, $\boldsymbol{s} \to^{+} \boldsymbol{s}'$ implies $\boldsymbol{s}' \in A$.

Attractors are hence just absorbing strongly connected components in the STG. An attractor A such that $|A| = 1$ is called a *steady state* (also named *point attractor*). We also denote with $|A|$ the *length* of attractor A.

3 Backward Boolean Equivalence

Our reduction method is based on the notion of backward equivalence, recast for BNs, which proved to be effective for reducing the dimensionality of ordinary differential equations [9,13] and chemical reaction networks [11,6,8]. Section 3.1 introduces *Backward Boolean Equivalence* (BBE), which is an equivalence relation on the variables of a BN, and use it to obtain a reduced BN. Section 3.2 provides an algorithm which iteratively compute the maximal BBE of a BN. Section 3.3 relates the properties of an original and BBE-reduced BN.

We fix a BN $B = (X, F)$, with $|X| = n$. We use R to denote equivalence relations on X and X_R for the induced partition.

3.1 Backward Boolean Equivalence and BN Reduction

We first introduce the notion of *constant* state on an equivalence relation R.

Definition 4. ***(Constant State)*** *A state* $\mathbf{s} \in \mathbb{B}^n$ *is* constant *on* R *if and only if* $\forall (x_i, x_j) \in R$ *it holds that* $s_{x_i} = s_{x_j}$.

Consider our running example and an equivalence relation R given by the partition $X_R = \{\{x_{Sp8}, x_{Fgf8}\}, \{x_{Pax6}\}, \{x_{Emx2}\}, \{x_{Coup\_tfi}\}\}$. The states constant on R are colored in purple in Fig. 3. For example, the state $\mathbf{s} = (1,0,1,1,1)$ is constant on R because $s_{Sp8} = s_{Fgf8}$ (the first and fourth positions of $\mathbf{s}$, respectively). On the contrary, $(1,0,1,0,1)$ is not constant on R.

We now define *Backward Boolean Equivalence (BBE).*

Definition 5. ***(Backward Boolean Equivalence)*** *Let* $B = (X, F)$ *be a BN,* X_R *a partition of the set* X *of variables, and* $C \in X_R$ *a class of the partition. A partition* X_R *is a* Backward Boolean Equivalence (BBE) *if and only if the following formula is valid:*

$$\Phi^{X_R} \equiv \left(\bigwedge_{\substack{C \in X_R \\ x, x' \in C}} \Big(x = x' \Big) \right) \longrightarrow \bigwedge_{\substack{C \in X_R \\ x, x' \in C}} \Big(f_x(X) = f_{x'}(X) \Big)$$

Φ^{X_R} says that if for all equivalence classes C the variables in C are equal, then the update functions of variables in the same equivalence class stay equal.

In other words, R is a BBE if and only if for all $\mathbf{s} \in \mathbb{B}^n$ constant on R it holds that $F(\mathbf{s})$ is constant on R. BBE is a relation where the update functions F preserve the "constant" property of states. The partition $X_R = \{\{x_{Sp8}, x_{Fgf8}\}, \{x_{Pax6}\}, \{x_{Emx2}\}, \{x_{Coup\_tfi}\}\}$ described above is indeed a BBE. This can be verified on the STG: all purple states (the constant ones) have outgoing transitions only towards purple states.

We now define the notion of BN reduced up to a BBE R. Each variable in the reduced BN represents one equivalence class in R. We denote by $f\{{}^{a}\!/_{b}\}$ the term arising by replacing each occurrence of b by a in the function f.

Definition 6. *The reduction of B up to R, denoted by B/R, is the BN (X_R, F_R) where $F_R = \{f_{x_C} : C \in X_R\}$, with $f_{x_C} = f_{x_k}\{^{x_{C'}}/_{x_i} : \forall C' \in X_R, \forall x_i \in C'\}$ for some $x_k \in C$.*

The definition above uses one variable per equivalence class, selects the update function of any variable in such class, and replaces all variables in it with a representative one per equivalence class. Fig. 4 shows the reduction of the cortical area development BN. We selected the update function of x_{Sp8} as the update function of the class-variable $x_{\{Fgf8,Sp8\}}$, and replaced every occurrence of x_{Sp8} and x_{Fgf8} with $x_{\{Fgf8,Sp8\}}$. The STG of such reduced BN is given in Fig. 3(b).

$$\begin{aligned}
x_{\{Fgf8,Sp8\}}(t+1) &= x_{\{Fgf8,Sp8\}}(t) \wedge \neg x_{\{Emx2\}}(t)\\
x_{\{Pax6\}}(t+1) &= \neg x_{\{Emx2\}}(t) \wedge x_{\{Fgf8,Sp8\}}(t) \wedge \neg x_{\{Coup\_tfi\}}(t)\\
x_{\{Emx2\}}(t+1) &= \neg x_{\{Fgf8,Sp8\}}(t) \wedge \neg x_{\{Pax6\}}(t) \wedge \neg x_{\{Fgf8,Sp8\}}(t) \wedge x_{\{Coup\_tfi\}}(t)\\
x_{\{Coup\_tfi\}}(t+1) &= \neg x_{\{Fgf8,Sp8\}}(t) \wedge \neg x_{\{Fgf8,Sp8\}}(t)
\end{aligned}$$

Fig. 4: The BBE-reducion of the cortical area development network of Fig. 2.

3.2 Computation of the maximal BBE

A crucial aspect of BBE is that it can be checked directly on a BN without requiring the generation of the STG. This is feasible by encoding the logical formula of Definition 5 into a logical SATisfiability problem [3]. A SAT solver has the ability to check the validity of such a logical formula by checking for the unsatisfiability of its negation ($sat(\neg\Phi^{X_R})$). A partition X_R is a BBE if and only if sat($\neg\Phi^{X_R}$) returns "unsatifiable", otherwise a counterexample (a witness) is returned, consisting of variables assignments that falsify Φ^{X_R}. Using counterexamples, it is possible to develop a partition refinement algorithm that computes the largest BBE that refines an initial partition.

The partition refinement algorithm is shown in Algorithm 1. Its input are a BN and an initial partition of its variables X. A *default* initial partition that leads to the maximal reduction consists of one block only, containing all variables. In general, the modeller may specify a different initial partition if some variables should not be merged together, placing them in different blocks. The output of the algorithm is the largest partition that is a BBE and refines the initial one.

We now explain how the algorithm works for input the cortical area development BN and the initial partition $X_R = \{\{x_{Fgf8}, x_{Emx2}, x_{Pax6}, x_{Sp8}, x_{Coup\_tfi}\}\}$.

Iteration 1. The algorithm enters the *while* loop, and the solver checks if Φ^{X_R} is valid. X_R is not a BBE, therefore the algorithm enters the second branch of the *if* statement. The solver gives an example satisfying $\neg\Phi^{X_R}$: $s = (s_{x_{Fgf8}}, s_{x_{Pax6}}, s_{x_{Emx2}}, s_{x_{Sp8}}, s_{x_{Coup\_tfi}})$ =$(0,0,0,0,0)$. Since $t = F(s) = (0,0,\ 0,0,1)$, the *for* loop partitions G into $X_{R_1} = \{\{x_{Fgf8},\ x_{Pax6},\ x_{Emx2}\ x_{Sp8}\}, \{x_{Coup\_tfi}\}\}$. The state $t = (0,0,0,0,1)$ is now constant on X_{R_1}.

Algorithm 1: Compute the maximal BBE that refines the initial partition X_R for a BN (X, F)

```
Result: maximal BBE $H$ that refines $X_R$
H ← $X_R$;
while true do
    if $\Phi^H$ is valid then
        return H ;
    else
        $\mathbf{s}$ ← get a state that satisfy $\neg\Phi^H$;
        $H' \leftarrow \emptyset$;
        for $C \in H$ do
            $C_0 = \{x_i \in C : f_{x_i}(\mathbf{s}) = 0\}$;
            $C_1 = \{x_i \in C : f_{x_i}(\mathbf{s}) = 1\}$;
            $H' = H' \cup \{C_1\} \cup \{C_0\}$;
        end
        $H \leftarrow H' \setminus \{\emptyset\}$;
    end
end
```

Iteration 2. The algorithm checks if $\Phi^{X_{R_1}}$ is valid (i.e. if X_{R_1} is a BBE). X_{R_1} is not a BBE. The algorithm gives a counterexample with $s = (0, 0, 0, 0, 1)$ and $t = F(s) = (0, 0, 1, 0, 1)$. The *for* loop refines X_{R_1} into $X_{R_2} = \{\{x_{Fgf8}, x_{Pax6}$ $x_{Sp8}\}, \{x_{Emx2}\}, \{x_{Coup\_tfi}\}\}$. X_{R_2} makes $t = (0, 0, 1, 0, 1)$ constant.

Iteration 3. The algorithm checks if G_2 is a BBE. The formula $\neg\Phi^{X_{R_2}}$ is satisfiable, so G_2 is not a BBE, and the solver provides an example with $s = (1, 1, 0, 1, 1)$ and $F(s) = (1, 0, 0, 1, 0)$. Hence, X_{R_2} is partitioned into $X_{R_3} = \{\{x_{Fgf8}, x_{Sp8}\}, \{x_{Pax6}\}$ $\{x_{Emx2}\}, \{x_{Coup\_tfi}\}\}$.

Iteration 4. The SAT solver proves that $\Phi^{X_{R_3}}$ is valid.

The number of iterations needed to reach a BBE depends on the counterexamples that the SAT solver provides. As for all partition-refinement algorithms, it can be easily shown that the number of iterations is bound by the number of variables. Each iteration requires to solve a SAT problem which is known to be NP-complete, however we show in Section 4 that we can easily scale to the largest models present in popular BN repositories.

We first show that given an initial partition there exists exactly one *largest* BBE that refines it. [1]

After that, we prove that Algorithm 1 indeed provides the maximal BBE that refines the initial one.

Theorem 1. *Let $BN = (X, F)$ and X_R a partition. There exists a unique maximal BBE H that refines X_R.*

Theorem 2. *Algorithm 1 computes the maximal BBE partition refining X_R.*

[1] All proofs are given in the extended version of this paper [1].

3.3 Relating Dynamics of Original and Reduced BNs

Given a BN B and a BBE R, $STG(B/R)$ can be seen as the subgraph of $STG(B)$ composed of all states of $STG(B)$ that are constant on R and their transitions. Of course, those states are transformed in $STG(B/R)$ by "collapsing" BBE-equivalent variables in the state representation. This can be seen by comparing the STG of the our running example (left part of Fig. 3) and of its reduction (right part of Fig. 3). The states (and transitions) of the STG of the reduced BN correspond to the purple states of the original STG.

Let B be a BN with n variables, $S \subseteq \mathbb{B}^n$ be the states of its STG, and R a BBE for B. We use $S_{|R}$ to denote the subset of S composed by all and only the states constant on R. With $STG(B)_{|R}$ we denote the subgraph of $STG(B)$ containing $S_{|R}$ and its transitions. Formally $STG(B)_{|R} = (S_{|R}, T_{|R})$, where $T_{|R} = T \cap (S_{|R} \times S_{|R})$.

The following lemma formalizes a fundamental property of $STG(B)_{|R}$, namely that all attractors of B containing states constant on R are preserved in $STG(B)_{|R}$.

Lemma 1. ***(Constant attractors)*** *Let $B(X, F)$ be a BN, R be a BBE, and A an attractor. If $A \cap S_{|R} \neq \emptyset$ then $A \subseteq S_{|R}$.*

We now define the bijective mapping $m_R : S_{|R} \leftrightarrow S_R$ induced by a BBE R, where S_R are the states of $STG(B/R)$, as follows: $m_R(\mathbf{s}) = (v_{C_1}, \ldots, v_{C_{|X/R|}})$ where $v_{C_j} = s_{x_i}$ for some $x_i \in C_j$. In words m_R bijectively maps each state of $STG(B)_{|R}$ to their compact representation in $STG(B/R)$. Indeed, $STG(B)_{|R}$ and $STG(B/R)$ are isomorphic, with m_R defining their (bijective) relation. We can show this through the following lemma.

Lemma 2. ***(Reduction isomorphism)*** *Let $B(X, F)$ be a BN and R be a BBE. Then, it holds*

1. *For all states $\boldsymbol{s} \in S_{|R}$ it holds $F_R(m_R(\boldsymbol{s})) = m_R(F(\boldsymbol{s}))$.*
2. *For all states $\boldsymbol{s} \in S_R$ it holds $F(m_R^{-1}(\boldsymbol{s})) = m_R^{-1}(F_R(\boldsymbol{s}))$.*

The previous Lemma ensures that BBE does not generates spurious trajectories or attractors in the reduced system. We can now state the main result of our approach, namely that the BBE reduction of a BN for a BBE R exactly preserves all attractors that are constant on R up to renaming with m_R.

Theorem 3. ***(Constant attractor preservation)*** *Let $B(X, F)$ be a BN, R a BBE, and A an attractor. If $A \cap S_{|R} \neq \emptyset$ then $m_R(A)$ is an attractor for B/R.*

4 Application to BNs from the Literature

We hereby apply BBE to BNs from the GINsim repository. Section 4.1 validates BBE on all models from the repository, while Section 4.2 studies the runtime speedups brought by BBE on attractor-based analysis of selected case studies, showing cases for which BBE makes the analysis feasible. Section 4.3 compares

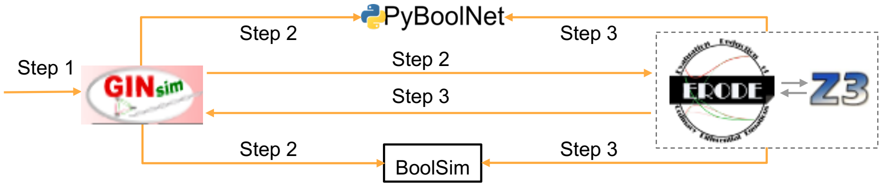


Fig. 5: BBE toolchain. (Step 1) We use GINsim [15] to access its model repository, and (Step 2) export it in the formats of the other tools in the toolchain to perform: STG generation (PyBoolNet [30]), attractor analysis (BoolSim [19]), and BBE reduction (ERODE [10]). (Step 3) We export the reduced models for analysis to PyBoolNet and BoolSim, or to GINsim.

BBE with the approach based on ODE encoding from [11], showing how such encoding leads to scalability issues and to the loss of reduction power.

The experiments have been made possible by a novel toolchain (Fig. 5) combining tools from the COLOMOTO initiative [33], and the reducer tool ERODE [10] which was extended here to support BBE-reduction. For Algorithm 1 we use the solver Z3 [17] which was already integrated in ERODE.

All experiments were conducted on a common laptop with an Intel Xeon(R) 2.80GHz and 32GB of RAM. We imposed an arbitrary timeout of 24 hours for each task, after which we terminated the analysis. We refer to these cases as *time-out*, while we use *out-of-memory* if a tool terminated with a memory error.

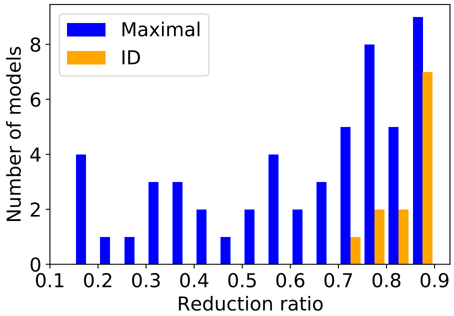
4.1 Large Scale Validation of BBE on BNs

We validate BBE on real-world BNs in terms of the number of BNs that can be reduced and the average reduction ratio.

Configuration. We conducted our investigation on the whole GINsim model repository which contains 85 networks: 29 are Boolean, and 56 are multivalued. In multivalued networks (MNs), some variables have more than 2 activation statuses, e.g. $\{0, 1, 2\}$. These models are automatically *booleanized* [18, 14] by GinSim when exporting in the input formats of the other tools in the tool-chain.

Most of the models in the repository have a specific structure [32] where a few variables are so-called *input variables*. These are variables whose update functions are either a stable function (e.g. $x(t+1) = 0$, $x(t+1) = 1$) or the identity function (e.g. $x(t+1) = x(t)$). These are named 'input' because their values are explicitly set by the modeler to perform experiments campaigns. We investigate two reduction scenarios relevant to input variables. In the first one, Algorithm 1 starts with initial partitions that lead to the *maximal reduction*, i.e. consisting of one block only. In the second scenario, we provide initial partitions that isolate inputs in singleton blocks. Therefore, we prevent their aggregation with other variables, and obtain reductions independent of the values of the input variables (we recall that BBE requires related variables to be initialized with same activation value). We call this case *input-distinguished (ID) reduction*.

Results. By using the maximal reduction setting, we obtained reductions on 61 of the 85 models, while we obtained ID reductions on 38 models. We summarize the reductions obtained for the two settings in Fig. 6, displaying the distribution of the reduction ratios $r_m = N_m/N$ and $r_i = N_i/N$, where N, N_m and N_i are the number of variables in the original BN, in the maximal BBE-reduction, and in the ID one, respectively. We also provide the average reduction ratios on the models, showing that it does not substantially change across Boolean or multivalued models. No reduction took more than 3 seconds.



Average reduction ratios		
	Maximal	*ID*
BNs	0.66	0.83
MNs	0.68	0.95
ALL	0.67	0.91

Fig. 6: (Left) Distribution of reduction ratios (reduced variables over original ones) on all models from the GINsim repository using the maximal and ID reduction strategy. Each bar counts the number of models with that reduction ratio, starting from 15% up to 90%, with step 5%. (Right) Average reduction ratios for Boolean, Multivalued and all models.

Interpretation. BBE reduced a large number of models (about 72%). In particular, this happened in 24 out of the 29 (83%) Boolean models and in 37 out of 56 (66%) multivalued networks. The average reduction ratio for the maximal and ID strategies are 0.67 and 0.91, respectively. For the former strategy, we get trivial reductions in 22 models wherein only input variables are related. In such trivial cases, the ID strategy does not lead to reduction. In other cases, the target variables of inputs (i.e. variables with incoming edges only from input variables considering the graphical representation of variables) appeared to be backward equivalent together with the input variables. This results in reductions with large equivalence classes consisting of input variables and their descendants. These are interesting reductions which get lost using the ID approach, as the input variables get isolated.

4.2 Attractor analysis of selected case studies

Hypothesis. We now investigate the fate of asymptotic dynamics after BBE-reduction, and test the computational efficiency in terms of time needed for attractor identification in the original and reduced models. We expect that BBE-reduction can be utilized to (i) gain fruitful insights into large BN models and (ii) to reduce the time needed for attractor identification.

Configuration. Our analysis focuses on three BNs from the GINsim repository. The first is the Mitogen-Activated Protein Kinases (MAPK) network [26] with 53 variables. The second refers to the survival signaling in large granular lymphocyte leukemia (T-LGL) [52] and contains 60 variables. The third is the merged Boolean model [40] of T-cell and Toll-like receptors (TCR-TLR5) which is the largest BN model in GINsim repository with 128 variables.

Results. The results of our analysis are summarized in Table 1 for the original, ID- and maximal-reduced BN. We present the number of variables (*size*) and of Attractors (*Attr.*), the time for attractor identification on the original model (*An. (s)*) and that for reduction plus attractor identification (*Red. + An. (s)*).

Original model				*ID reduction*			*Maximal reduction*		
	Size	*Attr.*	*An.(s)*	*Size*	*Attr.*	*Red.+An.(s)*	*Size*	*Attr.*	*Red.+An.(s)*
MAPK Network	53	40	16.50	46	40	15.33	39	17	3.49
T-LGL	60	264	123.43	57	264	86.84	52	6	3.49
TCR-TLR	128	—*Time Out*—		116	—*Time Out*—		95	2	31.29

Table 1: Reduction and attractor analysis on 3 selected case studies.

Interpretation. ID reduction preserves all attractors reachable from any combination of activation values for inputs. This is an immediate consequence of 2, Theorem 3 and the fact that number of attractors in the original and the ID reduced BN is the same (see Table 1). Maximal reduction might discard some attractors. We also note that, despite the limited reduction in terms of obtained number of variables, we have important analysis speed-ups, up to two orders of magnitude. Furthermore, the largest model could not be analyzed, while it took just 30 seconds to analyze its maximal reduction identifying 2 attractors.

4.3 Comparison with ODE-based approach from [11]

As discussed, BBE is based on the backward equivalence notion firstly provided for ordinary differential equations (ODEs), chemical reaction networks, and Markov chains [9, 11]. Notably, [11] shows how the notion for ODEs can be applied indirectly to BNs via an *odification* technique [49] to encode BNs as ODEs. Such odification transforms each BN variable into an ODE variable that takes values in the continuous interval [0,1]. The obtained ODEs preserve the attractors of the original BN because the equations of the two models coincide when all variables have value either 0 or 1. However, infinitely more states are added for the cases in which the variables do not have integer value.

Scalability. The technique from [11] has been proved able to handle models with millions of variables. Instead, the odification technique is particularly computationally intensive. Due to this, it failed on some models from the GINsim repository, including two from [22], namely *core_engine_budding_yeast_CC* and *coupled_budding_yeast_CC*, consisting of 39 and 50 variables, respectively. Instead, BBE could be applied in less than a second.

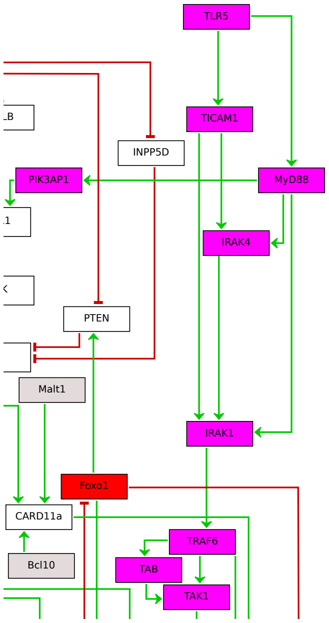


Fig. 7: Excerpt of GINsim's depict of TCR-TLR.

Reduction power. Another example is the *TCR-TLR* model from the previous section. In this case, both the ODE-based and BBE techniques succeeded. However, BBE led to better reductions due to the added non-integer states in the ODEs. Intuitively, the ODE-based technique *counts* incoming influences from equivalence classes of nodes, while BBE only checks whether at least one of such influence is present or not. Figure 7 shows an excerpt of the graphical representation of the model by GINsim. We use background colors of nodes to denote BBE equivalence classes (white denotes singleton classes). We see a large equivalence class of magenta species, 3 of which (*IRAK4*, *IRAK1*, and *TAK1*) receive two influences by magenta species, while the others receive only one. This differentiates the species in the ODE-based technique, keeping only the top four in the *magenta* block, while all the others end up in singleton blocks. We compare the original equations of *MyD88* and *IRAK4* which have 1 and 2 incoming influences each.

$$\begin{aligned} x_{MyD88}(t+1) &= x_{TLR5}(t) \\ x_{IRAK4}(t+1) &= (\neg x_{MyD88}(t) \land x_{TICAM1}(t)) \lor (x_{MyD88}(t)) \end{aligned}$$

We see that the two variables are BBE because their update functions depend only on the BBE-equivalent variables *TLR5* and *MyD88*, respectively. For *IRAK4*, the three variables in the update function are BBE. Therefore, they have same value allowing us to simplify the update function to just *MyD88*. The ODEs obtained for the 2 variables are, where $x'_{\_}$ denotes the derivative of $x_{\_}$:

$$\begin{aligned} x'_{MyD88} &= x_{TLR5} - x_{MyD88} \\ x'_{IRAK4} &= x_{MyD88} + x_{TICAM1} - x_{MyD88} \cdot x_{TICAM1} - x_{IRAK4} \end{aligned}$$

Given that all variables appearing in the equations are backward equivalent, the two equations coincide with the original ones when all variables have values either 0 or 1. However, they differ for non-integer values. For example, in case all variables have value 0.5, we get 0 for the former, and 0.25 for the latter.

5 Related Work

BN reduction techniques belong to three families according to their domain of reduction: (i) they reduce at syntactic level (i.e. the BN [34, 48, 39, 4, 32, 41, 51]), (ii) at semantic level (i.e. the STG [23, 2]), or (iii) they transform BNs to other formalisms like Petri Nets [16, 44] and ordinary differential equations [50] offering formalism-specific reductions. However, (semantic) STG-reduction does not solve the state space explosion whereas the transformation to other formalisms has several drawbacks as shown in Section 4.3.

Syntactic level reduction methods usually perform variable absorption [4, 34, 48, 41] at the BN. BN variables can get absorbed by the update functions of their target variables by replacing all occurrences of the absorbed variables with their update functions. This method was first investigated in [34] wherein update functions are represented as ordinary multivalued decision diagrams. The authors consider multivalued networks with updates being applied asynchronously and iteratively implement absorption. The process, despite preserving steady states in all synchronization schemas [48], might lead to loss of cycle attractors in the synchronous schema. However, absorption of variables might lead to introduction of new attractors in the asynchronous case, i.e., by reducing the number of variables the number of attractors can stay the same or increase (attractors can split or new attractors can appear).

A similar study [48] presents a reduction procedure and proves that it preserves steady states. This procedure includes two steps. The first refers to the deletion of links between variables on their network structure. Deletion of pseudo-influences is feasible by simplifying the Boolean expressions in update functions. The second step of the procedure refers to the absorption of variables like in [34].

The difference between studies [48], [34] is that [48] exploits Boolean algebra instead of multivalued decision diagrams to explain absorption. Moreover, they refer only to Boolean networks, and do not consider any update schema. In studies [34, 48, 41], self-regulated BN variables (i.e. variables with a self-loop in the graphical representation) can not be selected for absorption. The inability to absorb self-regulated variables is inherent in the implementation of absorption in contrast to our method where the restrictions are encoded by the user at the initial partition and self-regulated variables can be merged with other variables.

In [41] the authors presented a two step reduction algorithm. The first step includes the absorption of input variables with stable function and the second step the absorption of single mediator variables (variables with one incoming and outgoing edge in the signed interaction graph). The first step of the algorithm in [41] is equally useful and compatible with the first step of [48]. Moreover, if we combine the first steps of [48] and [41], we may achieve interesting reductions which exactly preserve all asymptotic dynamics.

The first steps of [48, 41] affect only a BN property called *stability*. Stability is the ability of a BN to end up to the same attractor when starting from slightly different initial conditions. In [4], the authors introduced the decimation procedure -a reduction procedure for synchronous BNs- to discuss how it affects stability. The crucial difference between decimation procedure and BBE-reduction is that the first was invented to study stability whereas the latter was invented to degrade state space explosion. The decimation procedure is summarized by the following four steps: (i) remove from every update functions the inputs that it does not depend on, (ii) find the constant value for variables with no inputs, (iii) propagate the constant values to other update functions and remove this variable from the system, and (iv) if a variable has become constant, repeat from step (i). The study also refers to leaf variables because their presence does not play any role in the asymptotic dynamics of a BN. However, both leaf and fixed-valued variables affect stability. Overall, the decimation procedure exactly preserves the asymptotic dynamics of the original model since it throws out only variables considered as asymptotically irrelevant.

6 Conclusion

We introduced an automatic reduction technique for synchronous Boolean Networks which preserves dynamics of interest. The modeller gets a reduced BN based on requirements expressed as an initial partition of variables. The reduced BN can recover a pure part of the original state space and its trajectories established by the reduction isomorphism. Notably, we draw connections between the STG of the original and that of the reduced BN through a rigorous mathematical framework. The dynamics preserved are those wherein collapsed variables have equal values.

We used our reduction technique to speed-up attractor identification. Despite that the length of the preserved attractors is consistent in the reduced model, some of them may get lost. In the future, we plan to study classes of initial partitions that preserve all attractors. We have shown the analysis speed-ups obtained for attractor identification as implemented in the tool BoolSim [24]. In the future we plan to perform a similar analysis on a recent attractor identification approach from [21].

Our method was implemented in ERODE [10], a freely available tool for reducing biological systems. Related *quantitative* techniques offered by ERODE have been recently validated on a large database of biological models [37, 38, 5]. In the future we plan to extend this analysis considering also BBE. We also plan to investigate whether BBE can be extended in order to be able to compare different models as done for its quantitative counterparts [7, 12].

Our method could be combined with most of the existing methods found in literature. Our prototype toolchain consists of several tools from the COLOMOTO interoperability initiative. We aim to incorporate our toolchain into the COLOMOTO Interactive Notebook [35], a unified environment to edit, execute, share, and reproduce analyses of qualitative models of biological networks.

Multivalued BNs, i.e. whose variables can take more than two activation values, are currently supported only via a *booleanization* technique [18, 14] that might hamper the interpretability of the reduced model. In future work we plan to generalize BBE to support directly multivalued networks.

References

1. Argyris, G., Lafuente, A.L., Tribastone, M., Tschaikowski, M., Vandin, A.: Reducing boolean networks with backward boolean equivalence - extended version (2021), https://arxiv.org/abs/2106.15476
2. Bérenguier, D., Chaouiya, C., Monteiro, P.T., Naldi, A., Remy, E., Thieffry, D., Tichit, L.: Dynamical modeling and analysis of large cellular regulatory networks. Chaos: An Interdisciplinary Journal of Nonlinear Science **23**(2), 025114 (2013)
3. Biere, A., Biere, A., Heule, M., van Maaren, H., Walsh, T.: Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications. IOS Press, NLD (2009)
4. Bilke, S., Sjunnesson, F.: Stability of the Kauffman model. Physical Review E **65**(1), 016129 (2001)
5. Cardelli, L., Perez-Verona, I.C., Tribastone, M., Tschaikowski, M., Vandin, A., Waizmann, T.: Exact maximal reduction of stochastic reaction networks by species lumping. Bioinformatics (2021). https://doi.org/10.1093/bioinformatics/btab081, https://doi.org/10.1093/bioinformatics/btab081
6. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Forward and backward bisimulations for chemical reaction networks. In: 26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015. pp. 226–239 (2015). https://doi.org/10.4230/LIPIcs.CONCUR.2015.226, https://doi.org/10.4230/LIPIcs.CONCUR.2015.226
7. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Comparing chemical reaction networks: A categorical and algorithmic perspective. In: Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016. pp. 485–494 (2016). https://doi.org/10.1145/2933575.2935318, https://doi.org/10.1145/2933575.2935318
8. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Efficient syntax-driven lumping of differential equations. In: Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings. pp. 93–111 (2016). https://doi.org/10.1007/978-3-662-49674-9_6, https://doi.org/10.1007/978-3-662-49674-9_6
9. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Symbolic computation of differential equivalences. In: Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016. pp. 137–150 (2016). https://doi.org/10.1145/2837614.2837649, https://doi.org/10.1145/2837614.2837649
10. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Erode: a tool for the evaluation and reduction of ordinary differential equations. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 310–328. Springer (2017)
11. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Maximal aggregation of polynomial dynamical systems. Proceedings of the National Academy of Sciences **114**(38), 10029–10034 (2017)
12. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Comparing chemical reaction networks: A categorical and algorithmic perspective. Theor.

Comput. Sci. **765**, 47–66 (2019). https://doi.org/10.1016/j.tcs.2017.12.018, https://doi.org/10.1016/j.tcs.2017.12.018

13. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Symbolic computation of differential equivalences. Theor. Comput. Sci. **777**, 132–154 (2019). https://doi.org/10.1016/j.tcs.2019.03.018, https://doi.org/10.1016/j.tcs.2019.03.018
14. Chaouiya, C., Bérenguier, D., Keating, S.M., Naldi, A., Van Iersel, M.P., Rodriguez, N., Dräger, A., Büchel, F., Cokelaer, T., Kowal, B., et al.: SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. BMC systems biology **7**(1), 1–15 (2013)
15. Chaouiya, C., Naldi, A., Thieffry, D.: Logical modelling of gene regulatory networks with ginsim. In: Bacterial Molecular Networks, pp. 463–479. Springer (2012)
16. Chaouiya, C., Remy, E., Thieffry, D.: Petri net modelling of biological regulatory networks. Journal of Discrete Algorithms **6**(2), 165–177 (2008)
17. De Moura, L., Bjørner, N.: Z3: An efficient smt solver. In: International conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 337–340. Springer (2008)
18. Delaplace, F., Ivanov, S.: Bisimilar booleanization of multivalued networks. BioSystems p. 104205 (2020)
19. Di Cara, A., Garg, A., De Micheli, G., Xenarios, I., Mendoza, L.: Dynamic simulation of regulatory networks using squad. BMC bioinformatics **8**(1), 462 (2007)
20. Drossel, B.: Random boolean networks. Reviews of nonlinear dynamics and complexity **1**, 69–110 (2008)
21. Dubrova, E., Teslenko, M.: A sat-based algorithm for finding attractors in synchronous boolean networks. IEEE/ACM transactions on computational biology and bioinformatics **8**(5), 1393–1399 (2011)
22. Fauré, A., Naldi, A., Lopez, F., Chaouiya, C., Ciliberto, A., Thieffry, D.: Modular logical modelling of the budding yeast cell cycle. Molecular bioSystems **5**, 1787–96 (2009 Dec 2009)
23. Figueiredo, D.: Relating bisimulations with attractors in boolean network models. In: International Conference on Algorithms for Computational Biology. pp. 17–25. Springer (2016)
24. Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., De Micheli, G.: Synchronous versus asynchronous modeling of gene regulatory networks. Bioinformatics **24**(17), 1917–1925 (07 2008). https://doi.org/10.1093/bioinformatics/btn336, https://doi.org/10.1093/bioinformatics/btn336
25. Giacomantonio, C.E., Goodhill, G.J.: A boolean model of the gene regulatory network underlying mammalian cortical area development. PLOS Computational Biology **6**(9), 1–13 (09 2010). https://doi.org/10.1371/journal.pcbi.1000936, https://doi.org/10.1371/journal.pcbi.1000936
26. Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perles, B., Thieffry, D.: Integrative modelling of the influence of mapk network on cancer cell fate decision. PLoS Comput Biol **9**(10), e1003286 (2013)
27. Hopfensitz, M., Müssel, C., Maucher, M., Kestler, H.A.: Attractors in boolean networks: a tutorial. Computational Statistics **28**(1), 19–36 (2013)
28. Kauffman, S.: Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology **22**(3), 437 – 467 (1969)
29. Klamt, S., Haus, U.U., Theis, F.: Hypergraphs and cellular networks. PLoS computational biology **5**(5) (2009)

30. Klarner, H., Streck, A., Siebert, H.: Pyboolnet: a python package for the generation, analysis and visualization of boolean networks. Bioinformatics **33**(5), 770–772 (2017)
31. Naldi, A., Berenguier, D., Fauré, A., Lopez, F., Thieffry, D., Chaouiya, C.: Logical modelling of regulatory networks with ginsim 2.3. Biosystems **97**(2), 134–139 (2009)
32. Naldi, A., Monteiro, P.T., Chaouiya, C.: Efficient handling of large signalling-regulatory networks by focusing on their core control. In: International Conference on Computational Methods in Systems Biology. pp. 288–306. Springer (2012)
33. Naldi, A., Monteiro, P.T., Müssel, C., for Logical Models, C., Tools, Kestler, H.A., Thieffry, D., Xenarios, I., Saez-Rodriguez, J., Helikar, T., Chaouiya, C.: Cooperative development of logical modelling standards and tools with colomoto. Bioinformatics **31**(7), 1154–1159 (2015)
34. Naldi, A., Remy, E., Thieffry, D., Chaouiya, C.: Dynamically consistent reduction of logical regulatory graphs. Theoretical Computer Science **412**(21), 2207–2218 (2011)
35. Naldi, A., Hernandez, C., Levy, N., Stoll, G., Monteiro, P.T., Chaouiya, C., Helikar, T., Zinovyev, A., Calzone, L., Cohen-Boulakia, S., Thieffry, D., Paulevé, L.: The colomoto interactive notebook: Accessible and reproducible computational analyses for qualitative biological networks. Frontiers in Physiology **9**, 680 (2018). https://doi.org/10.3389/fphys.2018.00680, https://www.frontiersin.org/article/10.3389/fphys.2018.00680
36. Paige, R., Tarjan, R.E.: Three partition refinement algorithms. SIAM Journal on Computing **16**(6), 973–989 (1987)
37. Pérez-Verona, I.C., Tribastone, M., Vandin, A.: A large-scale assessment of exact model reduction in the biomodels repository. In: Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings. pp. 248–265 (2019). https://doi.org/10.1007/978-3-030-31304-3_13, https://doi.org/10.1007/978-3-030-31304-3_13
38. Perez-Verona, I.C., Tribastone, M., Vandin, A.: A large-scale assessment of exact lumping of quantitative models in the biomodels repository. Theoretical Computer Science (2021). https://doi.org/https://doi.org/10.1016/j.tcs.2021.06.026, https://www.sciencedirect.com/science/article/pii/S0304397521003716
39. Richardson, K.A.: Simplifying boolean networks. Advances in Complex Systems **8**(04), 365–381 (2005)
40. Rodríguez-Jorge, O., Kempis-Calanis, L.A., Abou-Jaoudé, W., Gutiérrez-Reyna, D.Y., Hernandez, C., Ramirez-Pliego, O., Thomas-Chollier, M., Spicuglia, S., Santana, M.A., Thieffry, D.: Cooperation between t cell receptor and toll-like receptor 5 signaling for cd4+ t cell activation. Science signaling **12**(577) (2019)
41. Saadatpour, A., Albert, R., Reluga, T.C.: A reduction method for boolean network models proven to conserve attractors. SIAM Journal on Applied Dynamical Systems **12**(4), 1997–2011 (2013)
42. Schwab, J.D., Kühlwein, S.D., Ikonomi, N., Kühl, M., Kestler, H.A.: Concepts in boolean network modeling: What do they all mean? Computational and Structural Biotechnology Journal **18**, 571 – 582 (2020). https://doi.org/https://doi.org/10.1016/j.csbj.2020.03.001, http://www.sciencedirect.com/science/article/pii/S200103701930460X
43. Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W.: Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics **18**(2), 261–274 (2002)

44. Steggles, L.J., Banks, R., Shaw, O., Wipat, A.: Qualitatively modelling and analysing genetic regulatory networks: a petri net approach. Bioinformatics **23**(3), 336–343 (2007)
45. Su, C., Pang, J.: Sequential control of boolean networks with temporary and permanent perturbations. arXiv preprint arXiv:2004.07184 (2020)
46. Thomas, R.: Regulatory networks seen as asynchronous automata: a logical description. Journal of theoretical biology **153**(1), 1–23 (1991)
47. Thomas, R.: Kinetic logic: a Boolean approach to the analysis of complex regulatory systems: proceedings of the EMBO course "formal analysis of genetic regulation", held in Brussels, September 6–16, 1977, vol. 29. Springer Science & Business Media (2013)
48. Veliz-Cuba, A.: Reduction of boolean network models. Journal of theoretical biology **289**, 167–172 (2011)
49. Wittmann, D.M., Krumsiek, J., Saez-Rodriguez, J., Lauffenburger, D.A., Klamt, S., Theis, F.J.: Transforming boolean models to continuous models: methodology and application to T-cell receptor signaling. BMC Systems Biology **3**(1), 98 (2009). https://doi.org/10.1186/1752-0509-3-98
50. Wittmann, D.M., Krumsiek, J., Saez-Rodriguez, J., Lauffenburger, D.A., Klamt, S., Theis, F.J.: Transforming boolean models to continuous models: methodology and application to t-cell receptor signaling. BMC systems biology **3**(1), 98 (2009)
51. Zañudo, J.G.T., Albert, R.: An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. Chaos: An Interdisciplinary Journal of Nonlinear Science **23**(2), 025111 (2013). https://doi.org/10.1063/1.4809777, https://doi.org/10.1063/1.4809777
52. Zhang, R., Shah, M.V., Yang, J., Nyland, S.B., Liu, X., Yun, J.K., Albert, R., Loughran, T.P.: Network model of survival signaling in large granular lymphocyte leukemia. Proceedings of the National Academy of Sciences **105**(42), 16308–16313 (2008)