

RESEARCH

Open Access

Reducing Boolean networks with backward equivalence



Georgios A. Argyris¹, Alberto Lluch Lafuente¹, Mirco Tribastone², Max Tschaikowski³ and Andrea Vandin^{1,4*} 

From The 19th conference on Computational Methods in Systems Biology
Bordeaux, France. 22-24 September 2021. <https://cmsb2021.labri.fr/>

*Correspondence:
andrea.vandin@santannapisa.it

¹ Department of Applied
Mathematics and Computer
Science, Technical University
of Denmark, Lyngby, Denmark

² SysMA Unit, IMT School
for Advanced Studies, Lucca, Italy

³ Department of Computer
Science, University of Aalborg,
Aalborg, Denmark

⁴ Department of Excellence
EMbeDS and Institute
of Economics, Sant'Anna School
for Advanced Studies, Pisa, Italy

Abstract

Background: Boolean Networks (BNs) are a popular dynamical model in biology where the state of each component is represented by a variable taking binary values that express, for instance, activation/deactivation or high/low concentrations. Unfortunately, these models suffer from the state space explosion, i.e., there are exponentially many states in the number of BN variables, which hampers their analysis.

Results: We present Boolean Backward Equivalence (BBE), a novel reduction technique for BNs which collapses system variables that, if initialized with same value, maintain matching values in all states. A large-scale validation on 86 models from two online model repositories reveals that BBE is effective, since it is able to reduce more than 90% of the models. Furthermore, on such models we also show that BBE brings notable analysis speed-ups, both in terms of state space generation and steady-state analysis. In several cases, BBE allowed the analysis of models that were originally intractable due to the complexity. On two selected case studies, we show how one can tune the reduction power of BBE using model-specific information to preserve all dynamics of interest, and selectively exclude behavior that does not have biological relevance.

Conclusions: BBE complements existing reduction methods, preserving properties that other reduction methods fail to reproduce, and vice versa. BBE drops all and only the dynamics, including attractors, originating from states where BBE-equivalent variables have been initialized with different activation values. The remaining part of the dynamics is preserved exactly, including the length of the preserved attractors, and their reachability from given initial conditions, without adding any spurious behaviours. Given that BBE is a model-to-model reduction technique, it can be combined with further reduction methods for BNs.

Keywords: Boolean network, Model reduction, State-space generation, Attractors analysis, Partition refinement



Background

Boolean networks (BNs) are a popular model in systems biology where the dynamics is qualitatively associated with two levels. These may express, for instance, on/off behavior in gene regulation or high/low concentrations of molecular compounds [1]. In a BN, the state is defined as a vector of Boolean variables, each representing a distinct component of the system under consideration. The time evolution of each variable is governed by an update function, i.e., a Boolean expression that encodes how the other variables affect the change of state at each (discrete) time step [2–5]. In the main text we focus on synchronous BNs whereby the next state is obtained by applying all the update functions to the activation values of the current state. However, in the supplementary material we show how our approach can be applied also to BNs with partially asynchronous update schema (see, e.g., the *priority classes* supported by the popular tool GINsim [6] as described in [7]).

From a computational viewpoint, BNs are challenging to analyze. For example, the state space of the network, known as the state transition graph (STG), has exponential size in the number of variables. Thus, a full enumeration of the state space is possible only for networks of limited size. Another relevant type of analysis concerns the computation of attractors, i.e., those sets of states toward which the system tends to evolve and remain [8, 9]; these are often associated with biologically intelligible conditions of the system under study such as cell differentiation [3, 10]. Attractor identification is NP-hard [11] and, even if efficient tools have been developed [12], they do not scale well for large BNs.

These computational difficulties have motivated the development of reduction methods to ease BN analysis. Available techniques can be classified in three families according to the type of reduction: (i) by reasoning directly on the BN structure [4, 13–16]; (ii) by reducing the underlying STG [5, 17]; (iii) by transforming a BN into other formalisms for which specific reduction techniques are available [18, 19]. The latter two classes suffer two main limitations. First, STG-based reductions are still subject to state space explosion since they require the full enumeration of the state space to start with. Second, reductions via other formalisms may not be complete in the sense that the dynamics of the original BN and of the transformed model are not equivalent, hence some reductions may be missed (see Additional file 1).

In the case of reduction methods at the BN level, popular examples are based on the notion of *variable absorption*, proposed originally in [14, 15]. The main idea is that certain BN variables can get removed by replacing their occurrences with their update functions. This is based on the assumption that those variables evolve over time scales that justify that they can be updated first in the model. Other methods remove *output/leaf* variables [4, 13] (variables that do not appear in the update functions of other variables) or *frozen* ones (variables that stabilize to the same value after some iterations independently of the initial conditions) [16].

Here we present a complementary type of reduction method based on the computation of a partition of the variables in the BN, whereby the future dynamics of variables in a block of the partition are equal whenever they start from the same condition. This can be convenient, for instance, if one is interested in studying the dynamics due to simultaneous activation or deactivation of groups of variables [20] (see also the case studies

presented in the *Results and discussion* section). We call this kind of relation a *Boolean backward equivalence* (BBE) because it is defined analogously to the notion of backward bisimulation for Markov chains [21], more recently extended to chemical reaction networks [19, 22] and ordinary differential equations [23]. Recently, it has been shown how this backward notion can be given also for linear differential algebraic equations (DAE) [24, 25]. Using DAE terminology, such backward notion has been related to the preservation of invariant subspaces.

Every reduction technique comes with its own intuitive interpretation. For example, if we consider variable absorption mentioned above, it is intuitively based on the idea of fast/slow decomposition: it is biologically plausible to absorb a variable in the update function of another when the former fires faster than the latter. BBE is based on the following three orthogonal considerations:

- BBE allows the modeler to discover chains of variables that, under some initialization conditions, describe the same dynamics. This might be interesting, e.g., to estimate the quality of a model: large BBE reductions might signal excessive redundancy in the model.
- As mentioned above, the modeler might be interested only in dynamics where two or more variables have simultaneous (de)activation value (see, e.g., [20]). The T-LGL case study in section *Results and discussion* further discusses this.
- In [23], it has been shown that this backward notion corresponds to Cardelli's *emulation* [26] which enables to relate a complex model with a simpler one. Interestingly, [26] discusses how emulation can be given an evolutionary interpretation. In fact, an original model can express all the dynamics of the reduced model. In addition, the original model can also express all additional dynamics coming from states where variables related by emulation have different activation values (not permitted in the reduced model because variables related by emulation get collapsed in the same reduced one). Given this richer dynamics of the original model, Cardelli uses selected case studies in [26] to argue how the original model can be seen as an evolved version of the reduced one. We do not further investigate this aspect for BBE. However, given that BBE is based as well on the mentioned backward notions, it is not surprising that there exists a similar relation among the dynamics expressed by the original and reduced model (cf Fig. 1).

The criteria for a candidate partition of variables to be a BBE are encoded into a *satisfiability* problem over the expressions of the BN's update functions: we synthesise a Boolean expression involving BN variables and check whether there exists at least one combination of truth values for the variables that makes such expression true. This type of test can be effectively implemented using tools known as SAT solvers [27].

If a partition is a BBE, a reduced BN can be obtained by choosing and maintaining only a representative variable for each partition block, and renaming all variables in the remaining update functions with the representative one from their block. The STG of the reduced network exactly preserves the original dynamics for all states that have equal values across variables in the same block (Fig. 1). Importantly, however, the reduction method does not require the generation of the original STG, making it possible to

$$\begin{array}{l}
 x_1(t+1) = \neg x_3(t) \vee x_1(t) \\
 x_2(t+1) = x_1(t) \vee x_2(t) \vee \neg x_3(t) \\
 x_3(t+1) = x_2(t) \wedge \neg x_3(t)
 \end{array}
 \xrightarrow{x_1, x_2 : BBE}
 \begin{array}{l}
 x_{1,2}(t+1) = \neg x_3(t) \vee x_{1,2}(t) \\
 x_3(t+1) = x_{1,2}(t) \wedge \neg x_3(t)
 \end{array}$$



Fig. 1 Boolean backward equivalence shown on a simple example. (Top-left) BN with three variables denoted by x_1 , x_2 , and x_3 . (Bottom-left) The underlying STG. Each node is labelled by a vector that defines the state of each variable; a directed edge denotes a transition from a source state to a target state by a synchronous application of the update functions. States 110 and 111 form an attractor. (Top-right) Variables x_1 and x_2 can be shown to be BBE-equivalent by inspecting their update functions. If they have the same value in a state, i.e. $x_1(t) = x_2(t)$, then they will be equivalent for all successor states since $x_2(t+1) = x_1(t) \vee x_2(t) \vee \neg x_3(t) = x_1(t) \vee x_1(t) \vee \neg x_3(t) = x_1(t) \vee \neg x_3(t) = x_1(t+1)$. Based on this, a reduced BN can be obtained by considering a representative variable for each block and rewriting the corresponding update functions in terms of those representatives (here the representative variable is denoted by $x_{1,2}$). (Bottom-right) The underlying STG agrees with the original one on all states that have equal values for variables in the same block (purple nodes in bottom-left panel). Instead, any other state (i.e. those where variables in the same BBE block have different value), is removed. The criteria for BBE only involve checks for the update functions of the original model, such that the generation of original STG can be circumvented

obtain a reduced STG also from instances that would not be analyzable due to their massive size.

A crucial property satisfied by BBE is that there exists a *maximal* reduction for each BN, i.e., the coarsest BBE partition. This can be computed using a partition-refinement algorithm in a similar fashion as in Markov chains [28], reaction networks [22] and differential equations [23]. The algorithm essentially builds upon a fundamental result in computer science to prove equivalences in formal languages [29]. Given an initial partition of variables, the algorithm splits the blocks of the partition to compute its coarsest refinement that satisfies the BBE criteria. Thus, the maximal reduction is obtained when all variables are in the same unique block of the initial partition. However, the possibility of arbitrarily choosing the initial partition unlocks model-specific reduction queries that preserve the dynamics of user-defined variables. For example, in typical BN models of signaling pathways [30, 31], certain variables may represent the input signals to upstream components such as receptors. Formally, inputs may be detected because their update functions are constants that represent the values of such inputs. In this case, a possibly more biologically relevant initial partition may separate inputs from the other variables, obtaining *input-separated* (IS) reductions.

Our partition-refinement algorithm takes a polynomial number of steps as a function of the number of BN variables. At each iteration, it queries a SAT solver to check for the BBE criteria. If the query is satisfiable, i.e., the current partition is not a BBE, the returned assignment is used to split the current partition and perform another iteration; if the query is unsatisfiable, it returns the current partition as the coarsest BBE refinement of the initial one. We fully develop our approach in the *Methods* section and in Additional file 2.

Interestingly, although our algorithm is theoretically as complex as SAT solving, it behaves effectively in practice. Using a prototype implementation available within the software tool ERODE [32], we demonstrate its performance on a large-scale validation across 86 BN models from two well-known repositories [6, 33]. We show that almost all BNs can be reduced by BBE, providing speed-ups for the computation of STGs and attractors by more than three orders of magnitude. In some cases, BBE could render the analysis feasible in instances that originally issued out-of-memory errors or that were stopped after long time outs. This comes at the cost that part of the original dynamics is lost. In particular, in the STG we preserve all and only the states where variables within the same BBE-block have the same value, and transitions among them. From this, and from the properties of BBE, we also get that the method preserves all and only the attractors containing at least one preserved state. This confirms that BBE is complementary to existing reduction techniques for BNs. Indeed, in several areas of science and engineering, it is common to have reduction techniques that:

- Preserve all dynamics but might add spurious ones. An example is [14] which preserves all attractors but might create new spurious ones. These often come with the name of *over-approximations* (this is because, e.g., [14] might over-approximate the set of attractors of a model by computing a larger set containing all original ones, plus some spurious ones);
- Do not preserve all the dynamics, but guarantee to not add spurious ones, like BBE. These often come with the name of *under-approximations* (this is because, e.g., BBE might under-approximate the set of attractors of a model by computing a smaller set containing only original ones, but potentially not all).

These two families of techniques are not comparable. They might be jointly used to obtain upper-bounds (the case of [14]), and lower-bounds (the case of BBE) on the actual number of attractors in a model.

This paper extends the previous conference version [34]. All numerical experiments have been redesigned by adding an additional model repository, by performing a large-scale validation of the analysis speed-ups offered by BBE, and by considering a more recent and efficient tool for identification of attractors. We have also performed a new large-scale validation on randomly generated BNs. Finally, we have generalised the theory to support also BNs with partially asynchronous update schema, and we have added in Additional file 3 a new case study considering one such BN.

Results and discussion

In this section, we perform a large-scale validation of BBE. We first check its reduction power on published models from the literature, and then we demonstrate how it facilitates the analysis tasks of STG generation and attractor computation. In particular, we show how BBE brings important analysis speed-ups, both in terms of STG generation, and attractor analysis. In several cases, BBE enables the analysis of models that were originally intractable due to their complexity.

After this, we use two selected case studies to show how one can tune the reduction power of BBE to preserve or exclude specific dynamics of interest.

Toolchain. We implemented our method in ERODE [32], a freely available software for the modeling, analysis, and reduction of biological systems modelled in terms of BNs [34], differential equations [35], and chemical reaction networks [19]. ERODE integrates the SAT solver Z3 [36]. Thanks to importing/exporting functionalities, we let ERODE interact with the COLOMOTO Notebook [37, 38], which integrates several tools for modeling and analysis of BNs. STG generation is performed using the tool PyBoolNet [39], while attractor identification is performed using the SAT-based tool BNS [12].¹

Configuration. All experiments were conducted on a machine equipped with an Intel Xeon(R) 2.80 GHz processor and 32 GB RAM. We imposed an arbitrary timeout of 8 h for each task, after which we terminated the analysis. We refer to these cases as *time-out*, while we use *out-of-memory* if the execution issued a memory error.

We conducted our investigation using two model repositories: GINsim repository [6] (http://ginsim.org/models_repository), which contains 83 models, and the Bio-models repository [42] (<https://www.ebi.ac.uk/biomodels/>), which contains 24 models, obtaining overall 98 distinct models (9 appeared in both repositories). From these, we restricted only to models with input variables, obtaining 86 models. In other words, we considered about 92% of the models available in the two repositories. This selection was done to avoid favouring BBE: in BNs without inputs, IS initial partitions correspond to the maximal ones, which, as the name says, allow for the best possible BBE reduction of a model in terms of aggregation power. Part of these 86 models, 45, are multi-valued networks, i.e. logical models wherein some variables take more than two activation statuses, e.g., {0, 1, 2} for *low*, *medium*, or *high* concentration respectively (see, e.g., [43]). We transform such models in dynamically equivalent BNs by applying a so-called *booleanization* technique [44], supported by GINsim [40].

We consider two reduction scenarios relevant to input variables, using *maximal* and *input-separated* (IS) initial partitions like \mathcal{H}_0 in Eq. (1) and \mathcal{H}'_0 in Eq. (2), respectively. In Additional file 4 we perform a similar analysis on randomly generated BNs.

Large-scale validation

Large-scale validation: reduction power. We begin by addressing the reduction power of BBE. For this, we consider the *reduction ratios* (variables in the reduced BN over the variables in the original one) obtained on all models.

Figure 2 displays the reduction ratios for both the maximal and IS reductions. We observe that almost all models can be reduced by BBE, in particular 93% admit a maximal reduction, while 91% admit an IS one. The reduction ratios distribute almost uniformly from 0.15 to 1.00 (no reduction), with average reduction ratio of 0.70 and 0.77 for maximal and IS reductions, respectively. For most models, the maximal and IS reduction ratios do not change significantly, meaning that BBE is effective also when we prevent input variables from merging with internal variables. All detailed results of this analysis can be found in Additional file 5: Table S3.

¹ Among the tools available in the COLOMOTO notebook, we could have opted for GINsim [40] and BoolSim [41] for STG generation and attractor identification, respectively, for BNs with synchronous update schema. In both cases, we have opted for the tools with best performances according to preliminary experiments we conducted, not reported here.

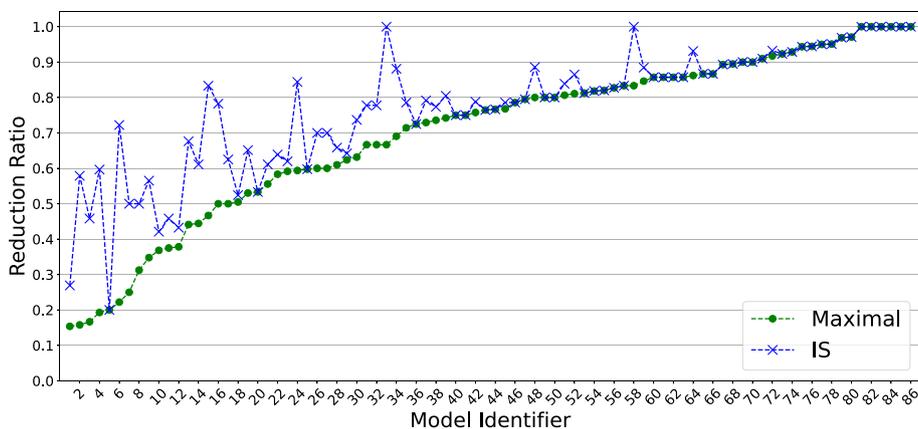


Fig. 2 Large-scale validation: reduction power. The x-axis provides model identifiers for the 86 considered models (only even ones are shown due to space constraints), while the y-axis refers to reduction ratios “reduced variables over original ones”. The green dots denote the reduction ratios, in increasing order, for maximal reductions. Using the same ordering, the blue crosses denote the reduction ratios for IS reductions. Only 6 models do not admit any BBE reduction, while two more models (33 and 58) do not admit IS reduction

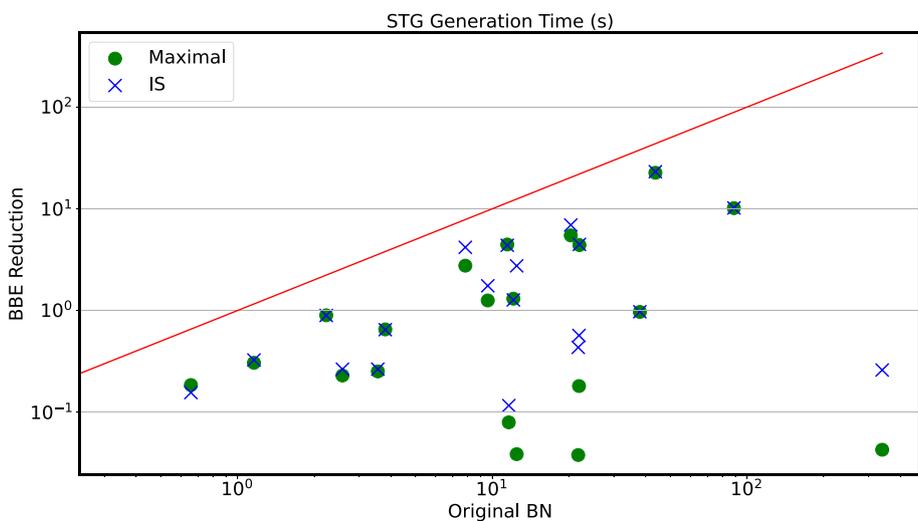


Fig. 3 Large-scale validation: STG generation speed-up. Comparison of STG generation between BBE reductions and original BNs. We omit models with 10 or fewer variables, where the runtimes are not particularly informative because STG generation is trivial. Furthermore, we omit models with more than 60 variables, where STG generation fails with *out-of-memory* for both the original models and their reductions. Overall, we obtain 33 models, from which here we focus only on the 20 ones for which the STG generation succeeded in both the original and reduced models, while Fig. 4 focuses on the remaining 13. The x-axis refers to the generation time for original models, while the y-axis refers to that for reduced models, using green circles and blue crosses for maximal and IS reductions, respectively. The runtimes are averaged over 3 runs

Large-scale validation: STG generation speed-up. We hereby demonstrate the speed-ups that BBE provides to STG generation on a selection of the considered models. Figure 3 focuses on the 20 models with more than 10 variables for which the STG generation succeeded in both the original and reduced models, while Fig. 4 focuses on the 13 ones where the STG generation failed on the original model and succeeded in the

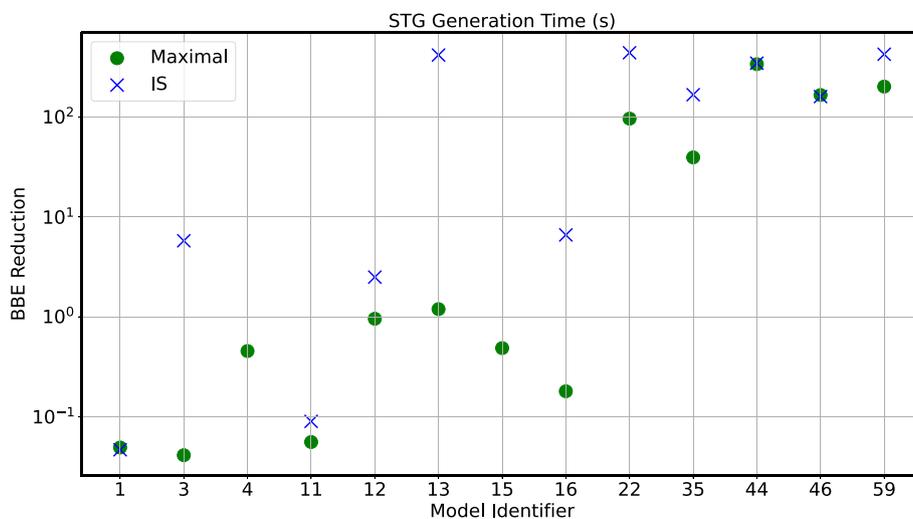


Fig. 4 Large-scale validation: STG generation speed-up. STG generation time for the maximal and the IS reductions. We consider the 13 models omitted in Fig. 3 because the STG generation failed for the original models. The x-axis refers to the model identifier from Fig. 2, while the y-axis refers to the generation time for the reduced models, using green circles and blue crosses for maximal and IS reductions, respectively. The runtimes are averaged over 3 runs

maximal or IS reduction. On the used machine, STG generation failed for models with 24 or more variables. Therefore, Fig. 3 focuses on models with less than 24 variables, while Fig. 4 focuses on models with 24 or more variables for which at least one reduction had less than 24 variables.

The red line in Fig. 3 marks the area where the reduction would not bring speed-ups. We can see that all points are below the line, with instances showing more than two orders of magnitude difference between the original and reduced runtimes. This proves that BBE can effectively lead to faster STG generation. All cases where the dots and crosses overlap refer to models where the two reductions coincide.

We now consider the 13 models in Fig. 4 where STG generation was not feasible for the original models. We note that the generation succeeded for all maximal reductions, while it failed for two IS ones. As denoted by the model identifiers in the x-axis, these are models 4 and 15 in Fig. 2, where the IS reductions have 34 and 25 variables, respectively. The largest runtime is 441 s for the IS reductions, and 338 s for the maximal ones. Detailed results are presented in Additional file 5: Table S4.

Large-scale validation: Attractor computation speed-up Fig. 5 studies the speed-ups that BBE provided to the computation of attractors on the models from Fig. 2. The plot has the same structure as Fig. 3. We observe that, in several cases, we have significant analysis speed-ups. In particular, we note how the dots and crosses spread to the right, due to original runtimes in the order of 10^3 s, while they hardly go up beyond 1 s for runtimes on reduced models. Furthermore, models 18 and 29 from Fig. 2 are omitted here because the analysis failed on the original models. Instead, the analysis of their maximal reductions required at most 0.15 s, and that of their IS reductions at most 2.5 s. Detailed results are given in Additional file 5: Table S5.

Large-scale validation: Interpretation. BBE can successfully reduce a large amount of models. For the original models, the state space explosion prevents full state space

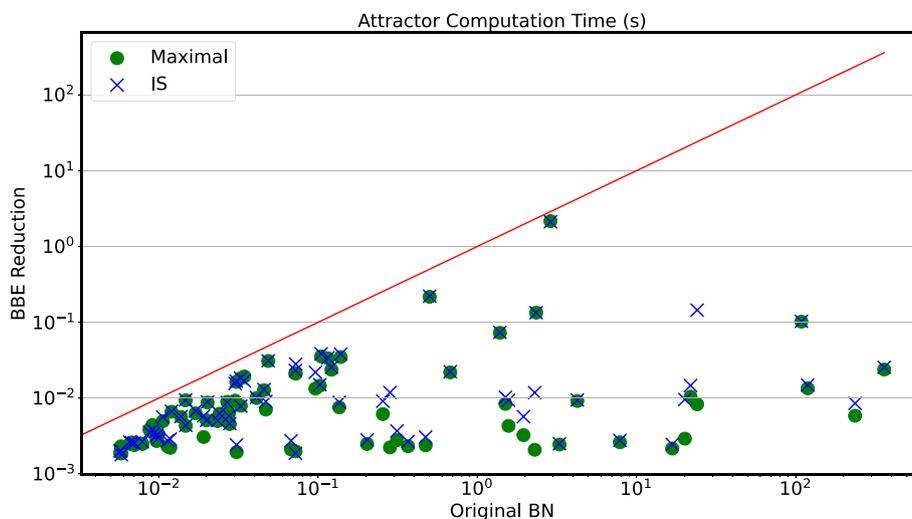


Fig. 5 Large-scale validation: Attractor computation time speed-up. Attractor computation time of the original models versus the one of maximal and IS reductions. Out of the 86 models from Fig. 2 we select the 78 admitting both maximal and IS reduction. The figure further omits models 18 and 29 from Fig. 2 for which the analysis failed for the original model due to *time-out*. The x-axis refers to the analysis time for original models, while the y-axis refers to that for reduced models, using green circles and blue crosses for maximal and IS reductions, respectively. The runtimes are averaged over 3 runs

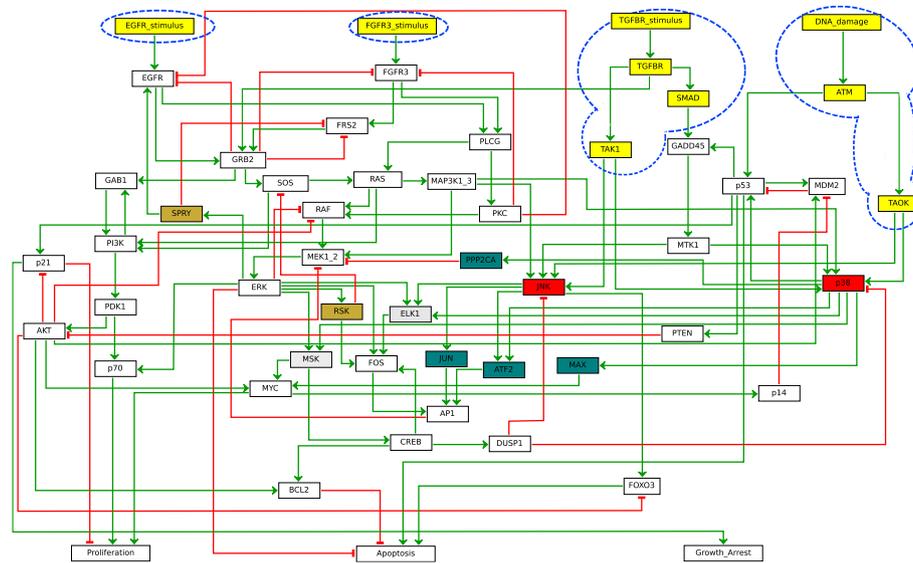
exploration in many cases, and hampers the identification of the attractors. This is mitigated in practice by BBE, with extreme cases where BBE made analysis feasible whereas the original models were intractable. As shown in Additional file 5: Table S5, part of the attractors are lost in the reduced models, namely those not involving constant states on the computed BBE (see Methods section). Table S5 shows cases like model 18 or 29, whose attractors could not be computed at all without BBE reduction. At the same time, the table shows that, by using the *default* IS or maximal initial partitions, a large part of the attractors might be lost (because, according to our theory, involve STG states where variables belonging to the same BBE block have different value). In Additional file 4: Fig. S7 we provide this information graphically for the BNs from the two repositories, and for randomly generated ones. This can be mitigated by devising *refined* initial partitions for the model and problem at hand. This is exemplified and discussed in greater detail in the next section where we show how a modeler can easily devise refined initial partitions that may allow to preserve more attractors, or drop those that are not of interest.

Case studies

In the previous part of this section we studied the aggregation power and the analysis speed-ups offered by BBE on 86 models from the literature. Here, instead, we use two selected case studies (MAPK, T-LGL) to show how one can tune, or *refine* the reduction power of BBE using model-specific information to preserve all dynamics of interest, and selectively exclude behavior that does not have biological relevance. Nevertheless, for completeness, we provide in Table 1 information on the analysis runtimes on all models (and their reductions) discussed in this section. We consider analysis runtimes on the models (*Original*), and on their IS (*IS*) and maximal (*Maximal*) reductions as done in

Table 1 Analysis runtimes (in seconds) for the models in section Case Studies

	MAPK				T-LGL			
	Original	IS	Maximal	Refined	Original	IS	Maximal	Refined
STG generation	Out-of-memory				Out-of-memory			
Attractor analysis	0.55	0.16	0.16	0.35	2.66	0.10	0.11	1.17

**Fig. 6** Graphical representation of the MAPK BN using GINsim. The background colors denote blocks of the maximal BBE (white background denotes singleton blocks). Instead, the blue dashed shapes denote blocks of the refined initial partition, vertical IS, where we omit the fifth large block containing all remaining nodes

the large-scale validation. Furthermore, we consider an additional reduction obtained using a refined initial partition discussed in the corresponding sections (*Refined*). STG generation failed on all models and reductions because they all have more than 24 variables. Indeed, we have previously discussed how, on the used machine, STG generation fails for models with 24 or more variables. Instead, attractor analysis succeeded on all models, with important speed-ups obtained for all reductions. For both models, the *IS* and *Maximal* cases have a particularly low analysis runtime. This is because, as we shall discuss next, several attractors are discarded in these reductions. Notably, despite the *Refined* reductions have speed-up factors of about two, as we shall see they preserve all attractors for MAPK, and all attractors of interest for T-LGL.

MAPK case study We consider a BN model for Mitogen-Activated Protein Kinase (MAPK) from [45]. The model consists of tightly interconnected signalling pathways involved in diverse cellular processes, such as cell cycle, survival, apoptosis and differentiation. The BN is depicted in Fig. 6. It contains 53 variables, 4 of which being inputs (*EGFR_stimulus*, *FGFR3_stimulus*, *TGFBR_stimulus*, and *DNA_damage*), and has 40 attractors.

MAPK: Maximal and IS reduction The maximal BBE reduction of this model has 39 variables. The discovered blocks are visualized in Fig. 6 using different background

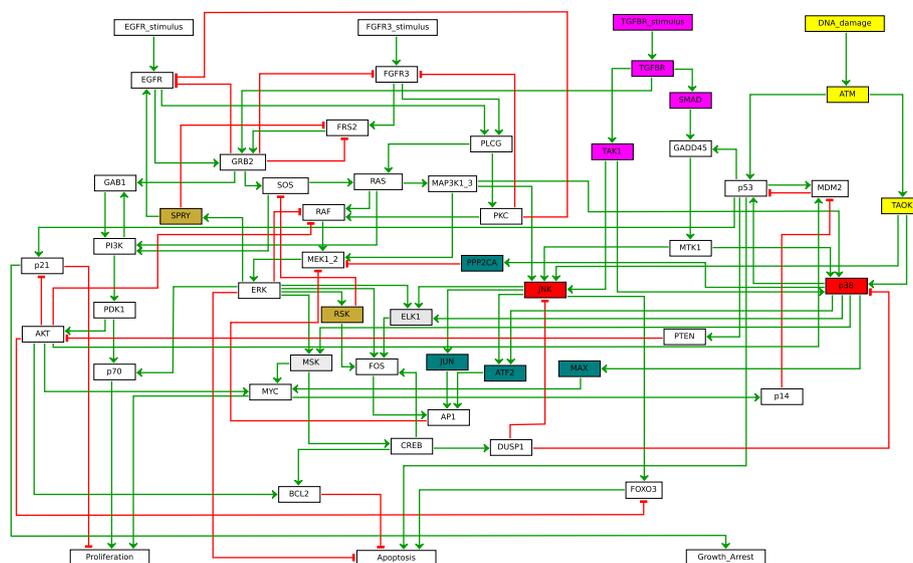


Fig. 7 Graphical representation of the MAPK BN. Background colors denote blocks of the BBE obtained using the refined initial partition (white background denotes singleton blocks)

colors. In particular, we note that the yellow block contains all inputs and five non-input variables, three related to *TGFBR_stimulus*, and two related to *DNA_damage*. Instead, the IS reduction has 41 variables, the only difference being that the block with inputs from the maximal reduction (Fig. 6) is split in three blocks: one for the inputs, one for the two non-input variables directly connected to the two right-most inputs, and one for the remaining non-input variables. In both cases, the reduced BNs have 17 attractors.

MAPK: Refined reduction with vertical IS We propose a third model-specific initial partition that considers inputs also *indirectly*. Intuitively, variables like *TGFBR* depend only on the value assigned to an input (*TGFBR_stimulus*). This reasoning can be iterated downward through the pathway, allowing to add also *TAK1*, and *SMAD*, until variables that depend on other (input) variables are met. In some sense, we can see *TGFBR*, *TAK1*, and *SMAD* as indirect inputs. This is because, in a few iterations the value assigned to the corresponding input will be propagated to them, and they will not change value anymore. In other words, we use a block per input, each containing the input and all non-input variables only positively affected by the input or by variables in the block. That way, we obtain an initial partition denoted by the blue dashed shapes in Fig. 6, plus an additional fifth block containing all other variables. The rationale is that a variable only affected by an input will have the same truth value of the input, therefore it can be considered as a sort of indirect input. The obtained BBE is depicted in Fig. 7. The reduced BN contains 42 variables and preserves all 40 attractors.

T-LGL case study We consider a BN model for T-LGL from [20]. It refers to the disease T-LGL leukemia which features a clonal expansion of antigen-primed, competent, cytotoxic T lymphocytes (CTL). This BN is a signalling pathway, constructed empirically through extensive literature review, and determines the survival of CTL.

equal values. These studies [4, 13] additionally remove the output [4] variables (also called leaf variables [13]). However, output variables sometimes are used to denote different “responses” by the modelled system [4, 30], therefore their removal might not always be appropriate.

In variable absorption [14, 15], the main assumption is that there are variables that are updated faster than others, therefore one class of variables can be assumed to be constant and absorbed if focusing on the dynamics of the other class. Unlike BBE, this can only increase the number of attractors. In particular, variable absorption preserves exactly all steady states (single-state attractors), while it might change the length of other attractors. Furthermore, new spurious attractors might be added. Instead, BBE might decrease the number of attractors (it discards all and only the attractors involving states where BBE-equivalent variables have different activation values), but all preserved attractors are preserved exactly, including their length and reachability from (preserved) initial states, and no spurious ones are added. Regarding other relevant work, in [16], the authors identify variables that have the same value in attractors only, but, differently from BBE, might behave differently in other states of the STG.

We validated BBE on 86 BNs from two model repositories, providing reductions and analysis speed-ups in almost all cases. In some, BBE enabled the analysis of models which would be otherwise intractable. There were also instances for which the reduced model could not be analyzed. This calls for further research into more aggressive reductions; for example, in its current implementation multi-valued BNs are first translated into ordinary BNs, but this causes a blow-up in the number of variables. It is worth investigating approaches that circumvent the intermediate translation to reduce dimensionality. Another area of research concerns the different semantic interpretations of a BN. Currently, BBE supports BN with synchronous and partially asynchronous updates; we plan to investigate variants of BBE for probabilistic BNs.

Methods

Here we explain the key steps of the reduction procedure on the BN in Fig. 9. Its customary graphical representation allows one to distinguish different kinds of variables depending on whether they appear in the update functions of other variables (as indicated by the green arrows). In the example, *TLR5* can be interpreted as an input because its state remains constant and unaffected by other variables. Inputs, which often denote external stimuli [4], are explicitly set by the modeler to perform experiment campaigns. Conversely, *IRAK4* and *PIK3AP1* can be considered output variables because they do not appear in the update functions of other variables.

Step 1: initial partition. Our reduction algorithm starts with the specification of an initial partition of variables. The idea behind initial partitions is that the modeler can force our algorithm to not collapse given variables, by placing them in different initial blocks. In the case studies presented in the *Results and discussion* section we see examples of user-specified initial partitions enabling analyses of interest on the considered models. This is how initial partitions shall be used, devising case-by-case useful ones. In order to favour a systematic large-scale validation of our approach, here we consider two examples of initial partitions whose computation can be easily automated: the *maximal* partition, where all variables are placed in the same block; and the *input-separated* (IS) one,

$$\begin{aligned}
 x_{TLR5}(t+1) &= f_{x_{TLR5}} = x_{TLR5}(t) \\
 x_{TICAM1}(t+1) &= f_{x_{TICAM1}} = x_{TLR5}(t) \\
 x_{MyD88}(t+1) &= f_{x_{MyD88}} = x_{TLR5}(t) \\
 x_{IRAK4}(t+1) &= f_{x_{IRAK4}} = x_{MyD88}(t) \vee x_{TICAM1}(t) \\
 x_{PIK3AP1}(t+1) &= f_{x_{PIK3AP1}} = x_{MyD88}(t)
 \end{aligned}$$

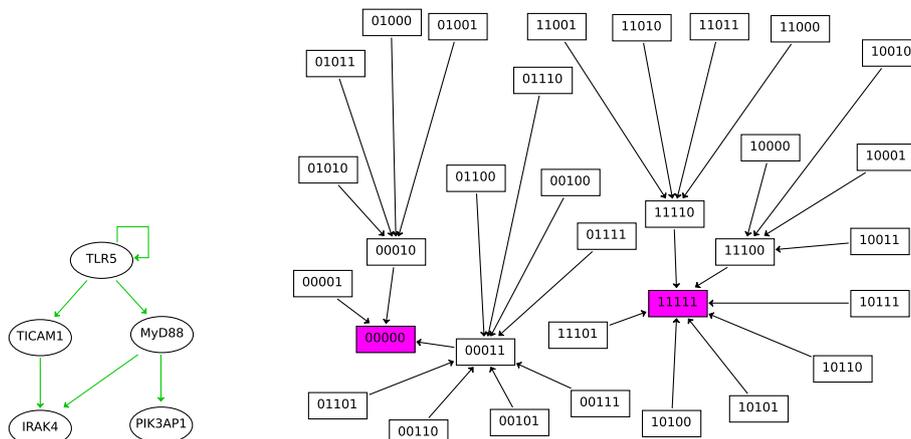


Fig. 9 Excerpt of the BN from [30]. It refers to the receptor *TLR5* and its signalling to the four following genes: *TICAM1*, *MyD88*, *IRAK4*, *PIK3AP1*. When a virus infects an organism, the receptor *TLR5* receives the relevant antigen stimuli becoming active (the value of x_{TLR5} turns from 0 to 1), and the signal is subsequently propagated to the other connected genes. (Top) The update functions of the BN. (Bottom-left) Variables are commonly depicted as nodes in a network while directed links represent influences between them. A directed link from a source variable to a target variable denotes that the source variable exists in the update function of the target variable. (Bottom-right) The corresponding STG, where we use purple to denote attractors

where the inputs are separated from the other variables, i.e., we use an initial partition with two blocks, one for input variables and one for the other variables.² In the example, these are respectively given by the partitions:

$$\mathcal{H}_0 = \{ \{x_{TLR5}, x_{TICAM1}, x_{MyD88}, x_{IRAK4}, x_{PIK3AP1}\} \} \tag{1}$$

and

$$\mathcal{H}'_0 = \{ \{x_{TLR5}\}, \{x_{TICAM1}, x_{MyD88}, x_{IRAK4}, x_{PIK3AP1}\} \}. \tag{2}$$

Iterative step: splitting by the BBE condition. At every iteration, the algorithm checks the BBE condition on the current partition. Formally, BBE is defined as a partition X of variables that renders the following formula valid:

$$\Phi^{\mathcal{H}} \equiv \left(\bigwedge_{\substack{H_i \in \mathcal{H} \\ x, x' \in H_i}} (x = x') \right) \longrightarrow \bigwedge_{\substack{H_i \in \mathcal{H} \\ x, x' \in H_i}} (f_x = f_{x'}) \tag{3}$$

² We refer to [34] for a third example of initial partition, *input-distinguished* where inputs were further separated from each other. As exemplified in the case studies in the *Results and discussion* section, initial partitions should be defined by the modeler depending on the model at hand and on the properties to be studied. We discuss the maximal and IS partitions here to enable the large-scale validation of BBE discussed in the same section.

This is a Boolean formula for: *whenever all variables in the same block have same value, they will not be distinguished in the next state*. In other words, $\Phi^{\mathcal{H}}$ says that if for all partition blocks H_i the variables in H_i are equal, then the evaluations of update functions of variables in the same block stay equal. A SAT solver can determine if $\Phi^{\mathcal{H}}$ is valid by checking the unsatisfiability of its negation. For example, given the \mathcal{H}'_0 partition in Eq. 2, one can obtain that $\neg\Phi^{\mathcal{H}'_0}$ is satisfiable (i.e., \mathcal{H}'_0 is not a BBE) because there exists the assignment s given by

$$s = (s_{x_{TLR5}}, s_{x_{TICAM1}}, s_{x_{MyD88}}, s_{x_{IRAK4}}, s_{x_{PIK3AP1}}) = (1, 0, 0, 0, 0)$$

for which, as it can be seen in the STG of Fig. 9, the next state s' is

$$s' = (s'_{x_{TLR5}}, s'_{x_{TICAM1}}, s'_{x_{MyD88}}, s'_{x_{IRAK4}}, s'_{x_{PIK3AP1}}) = (1, 1, 1, 0, 0).$$

This assignment proves that variables x_{TICAM1} , x_{MyD88} , x_{IRAK4} , and $x_{PIK3AP1}$ cannot belong to the same block of a partition that satisfies the BBE criteria because despite having the same value (0) in the source state s , they differ in the target state s' . In addition, the assignment s suggests to split that block into two sub-blocks for which that assignment does not disprove the BBE condition: x_{TICAM1} and x_{MyD88} have same value in s' , as well as x_{IRAK4} and $x_{PIK3AP1}$. Thus the algorithm will perform a new iteration with the refined partition

$$\mathcal{H}'_1 = \{\{x_{TLR5}\}, \{x_{TICAM1}, x_{MyD88}\}, \{x_{IRAK4}, x_{PIK3AP1}\}\}.$$

With this, $\neg\Phi^{\mathcal{H}'_1}$ is unsatisfiable, implying that \mathcal{H}'_1 is a BBE partition. In Theorem 2 from Additional file 2, we prove that this algorithm returns, for any initial partition, its unique coarsest refinement that satisfies the BBE condition (3). Overall, the algorithm takes at most n steps, where n is the number of BN variables; at every step, it iterates through the provided SAT assignment, if any is provided, to perform the splitting. Thus, overall the algorithm is as hard as SAT solving; however, the numerical evaluation presented in the *Results and discussion* section will show how it can effectively tackle BN models from the literature.

BBE properties As discussed in Fig. 1, given a BBE it is possible to construct a reduced BN where each variable represents a partition block (Proposition 4 from Additional file 2). The STG of the reduced BN agrees with the original STG on all, and only on, states that are *constant* on the partition, i.e., whose variables in the same block have the same value (Proposition 4 from Additional file 2). The reduction also preserves any attractor of the original BN which contains at least one state that is constant on the partition (Theorem 5 from Additional file 2). Thus, in particular the reduced BN maintains the exact length of the attractors that are preserved without introducing spurious dynamical behavior. Instead, all states non constant on the partition are dropped, as well as all attractors not containing any state constant on the BBE partition.

We use two examples to better explain the exact preservation of part of the attractors. Considering preserved attractors, we have seen in Fig. 1 that the two-states attractor of the original model (Fig. 1 bottom-left) is preserved in a two-states attractor in the reduced model (Fig. 1 bottom-right). This is the case for any preserved attractor; the number of states is preserved.

$$\begin{array}{lcl}
 x_1(t+1) & = & (x_1(t) \wedge \neg x_2(t)) \vee x_3(t) \\
 x_2(t+1) & = & (\neg x_1(t) \wedge x_2(t)) \vee x_3(t) \\
 x_3(t+1) & = & x_1(t) \wedge x_3(t)
 \end{array}
 \xrightarrow{x_1, x_2 : BBE}
 \begin{array}{lcl}
 x_{1,2}(t+1) & = & x_3(t) \\
 x_3(t+1) & = & x_{1,2}(t) \wedge x_3(t)
 \end{array}$$

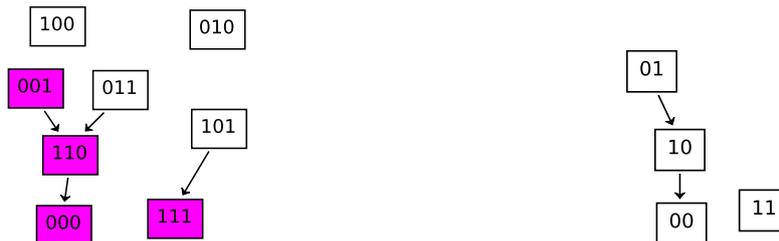


Fig. 10 Boolean backward equivalence shown on a simple example: not all attractors are preserved. (Top-left) BN with three variables denoted by x_1 , x_2 , and x_3 . (Bottom-left) The underlying STG. The model has 4 steady-state attractors (nodes 100, 010, 000, and 111). Two have same activation values for x_1 and x_2 (000, and 111), two have not. (Top-right) Variables x_1 and x_2 can be shown to be BBE-equivalent by inspecting their update functions. If they have the same value in a state, i.e. $x_1(t) = x_2(t)$, then they will be equivalent for all successor states. Based on this, a reduced BN can be obtained by considering a representative variable for each block and rewriting the corresponding update functions in terms of those representatives (here the representative variable is denoted by $x_{1,2}$). (Bottom-right) The underlying STG agrees with the original one on all states that have equal values for variables in the same block (purple nodes in bottom-left panel). Notably, the two attractors having same activation value for x_1 and x_2 are preserved, while the other two are dropped, as expected by our theory

As regards attractors that are not preserved, we provide in Fig. 10 (top-left) a simple BN with 3 variables (x_1 , x_2 , and x_3) and 4 attractors (steady-states, Fig. 10, bottom-left). Figure 10 (top-right) shows a BBE reduction of the model where x_1 and x_2 get collapsed. We can see in Fig. 10 (bottom-right) that 2 attractors are preserved in the BBE reduction, while the other 2 attractors belong to the part of the STG that is not preserved, and therefore are not present in the reduced BN. In particular, according to our theory, the two attractors where x_1 and x_2 are both 1 or both 0 are preserved. Instead, the other two attractors have different values for x_1 and x_2 , and therefore are not preserved.

Partially asynchronous BNs In Additional file 2, we show how BBE can also be applied to partially asynchronous BNs. Here, we equip a BN with a partition \mathcal{K} of its variables that we name *synchronization partition*. A new state is obtained by selecting one of the blocks K of \mathcal{K} , and then applying of the update functions of the variables in K only. The activation values of the other variables are not modified. Notably, this synchronization schema is supported, e.g., by popular BN analysis tools like GINsim [6] under the name of *priority classes* [7].³ In particular, BBE can be applied to such BNs with the caveat that the initial partition must be \mathcal{K} or refinements of it. In Additional file 3 we apply BBE to a BN with partially asynchronous update schema.

Abbreviations

BN	Boolean network
BBE	Boolean backward equivalence
STG	State transition graph

³ The dynamics of BNs considered in this paper are less general than those offered by GINsim using priority classes, as they also further allow to assign different priorities to the classes, and to update the variables within them asynchronously.

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s12859-023-05326-9>.

Additional file 1: Comparison with encoding-based reductions.

Additional file 2: Technical Results.

Additional file 3: Application of BBE to a BN with partially asynchronous schema.

Additional file 4: An application of BBE to randomly generated Boolean Networks.

Additional file 5: Tables with detailed results from large-scale experiments.

Acknowledgements

We thank reviewers of the original CMSB 2021 submission and of this submission for their help in improving the paper. We also thank the CMSB 2021 attendants for their comments and suggestions on the work. We thank Laure Bally-Cuif, author of [46], for her fruitful information on the modelling approach helping us in the case study in Additional file 3.

Author Contributions

AL, MiT, MaT, and AV conceived the study. GA, AL, MiT, MaT, and AV designed, developed and optimised the methods. GA, AL, MiT, MaT, and AV planned the experiments. GA and AV conducted the experiments and benchmarks. GA, AL, MiT, MaT, and AV wrote and revised the manuscript. All authors read and approved the final manuscript.

About this supplement

This article has been published as part of BMC Bioinformatics Volume 24 Supplement 1, 2023: Special Issue of the 19th International Conference on Computational Methods in Systems Biology. The full contents of the supplement are available online at <https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-24-supplement-1>.

Funding

Partially supported by the DFF project REDUCTO 9040-00224B, the Poul Due Jensen Grant 883901, the Villum Investigator Grant S405, and the PRIN project SEDUCE 2017TWRCNB. The funders had no direct role in the design of the study, collection, analysis and interpretation of the data, or writing of the manuscript. This publication was produced with the co-funding European Union - Next Generation EU, in the context of The National Recovery and Resilience Plan, Investment 1.5 Ecosystems of Innovation, Project Tuscany Health Ecosystem (THE), CUP: B83C22003920001.

Availability of data and materials

Material to replicate the large-scale experiments, including all models, are available here: https://www.erode.eu/models/BMCBioInf_CMSB2021.zip

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 15 March 2022 Accepted: 5 May 2023

Published online: 23 May 2023

References

1. Kauffman SA. Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol.* 1969;22(3):437–67.
2. Wang R-S, Saadatpour A, Albert R. Boolean modeling in systems biology: an overview of methodology and applications. *Phys Biol.* 2012;9(5):055001.
3. Azpeitia E, Benítez M, Vega I, Villarreal C, Alvarez-Buylla ER. Single-cell and coupled GRN models of cell patterning in the *Arabidopsis thaliana* root stem cell niche. *BMC Syst Biol.* 2010;4(1):1–19.
4. Naldi A, Monteiro PT, Chaouiya C. Efficient handling of large signalling-regulatory networks by focusing on their core control. In: International conference on computational methods in systems biology. Springer; 2012. p. 288–306.
5. Bérenquier D, Chaouiya C, Monteiro PT, Naldi A, Remy E, Thieffry D, Tichit L. Dynamical modeling and analysis of large cellular regulatory networks. *Chaos Interdiscip J Nonlinear Sci.* 2013;23(2):025114.
6. Naldi A, Bérenquier D, Fauré A, Lopez F, Thieffry D, Chaouiya C. Logical modelling of regulatory networks with GINsim 2.3. *Biosystems.* 2009;97(2):134–9.
7. Fauré A, Naldi A, Chaouiya C, Thieffry D. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics.* 2006;22(14):124–31.
8. Schwab JD, Kühlwein SD, Ikonomi N, Köhl M, Kestler HA. Concepts in Boolean network modeling: What do they all mean? *Comput Struct Biotechnol J.* 2020;18:571–82. <https://doi.org/10.1016/j.csbj.2020.03.001>.

9. Hopfensitz M, Müssel C, Maucher M, Kestler HA. Attractors in Boolean networks: a tutorial. *Comput Stat*. 2013;28(1):19–36.
10. Drossel B. Random Boolean networks. *Rev Nonlinear Dyn Complex*. 2008;1:69–110.
11. Akutsu T, Kuhara S, Maruyama O, Miyano S. A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions. *Genome Inform*. 1998;9:151–60.
12. Dubrova E, Teslenko M. A sat-based algorithm for finding attractors in synchronous Boolean networks. *IEEE/ACM Trans Comput Biol Bioinform*. 2011;8(5):1393–9.
13. Bilke S, Sjunnesson F. Stability of the Kauffman model. *Phys Rev E*. 2001;65(1):016129.
14. Naldi A, Remy E, Thieffry D, Chaouiya C. Dynamically consistent reduction of logical regulatory graphs. *Theor Comput Sci*. 2011;412(21):2207–18.
15. Veliz-Cuba A. Reduction of Boolean network models. *J Theor Biol*. 2011;289:167–72.
16. Richardson KA. Simplifying Boolean networks. *Adv Complex Syst*. 2005;8(04):365–81.
17. Figueiredo D. Relating bisimulations with attractors in Boolean network models. In: *International conference on algorithms for computational biology*. Springer; 2016. p. 17–25.
18. Zañudo JGT, Albert R. An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos Interdiscip J Nonlinear Sci*. 2013;23(2):025111. <https://doi.org/10.1063/1.4809777>.
19. Cardelli L, Tribastone M, Tschaikowski M, Vandin A. Maximal aggregation of polynomial dynamical systems. *Proc Natl Acad Sci*. 2017;114(38):10029–34.
20. Zhang R, Shah MV, Yang J, Nyland SB, Liu X, Yun JK, Albert R, Loughran TP. Network model of survival signaling in large granular lymphocyte leukemia. *Proc Natl Acad Sci*. 2008;105(42):16308–13.
21. Sproston J, Donatelli S. Backward bisimulation in Markov chain model checking. *Softw Eng IEEE Trans*. 2006;32(8):531–46. <https://doi.org/10.1109/TSE.2006.74>.
22. 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 14, 2015*. 2015. p. 226–239. <https://doi.org/10.4230/LIPLCS.CONCUR.2015.226>.
23. Cardelli L, Tribastone M, Tschaikowski M, Vandin A. Symbolic computation of differential equivalences. *ACM SIGPLAN Not*. 2016;51(1):137–50.
24. Tognazzi S, Tribastone M, Tschaikowski M, Vandin A. Differential equivalence for linear differential algebraic equations. *IEEE Trans Autom Control*. 2022;67(7):3484–93. <https://doi.org/10.1109/TAC.2021.3108530>.
25. Tognazzi S, Tribastone M, Tschaikowski M, Vandin A. Backward invariance for linear differential algebraic equations. In: *2018 IEEE conference on decision and control (CDC)*. 2018. p. 3771–3776. <https://doi.org/10.1109/CDC.2018.8619710>.
26. Cardelli L. Morphisms of reaction networks that couple structure to function. *BMC Syst Biol*. 2014;8(1):84.
27. 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.
28. Valmari A, Franceschinis G. Simple $O(m \log n)$ time markov chain lumping. In: *Tools and algorithms for the construction and analysis of systems, 16th international conference, TACAS 2010, held as part of the joint European conferences on theory and practice of software, ETAPS 2010, Paphos, Cyprus, March 20–28, 2010. Proceedings*. 2010. p. 38–52. https://doi.org/10.1007/978-3-642-12002-2_4.
29. Paige R, Tarjan R. Three partition refinement algorithms. *SIAM J Comput*. 1987;16(6):973–89. <https://doi.org/10.1137/0216062>.
30. Rodríguez-Jorge O, Kempis-Calanis LA, Abou-Jaoudé W, Gutiérrez-Reyna DY, Hernandez C, Ramirez-Pliego O, Thomas-Chollier M, Spicuglia S, Santana MA, Thieffry D. Cooperation between T cell receptor and toll-like receptor 5 signaling for CD4+ T cell activation. *Sci Signal*. 2019. <https://doi.org/10.1126/scisignal.aar3641>.
31. Klamt S, Saez-Rodríguez J, Lindquist JA, Simeoni L, Gilles ED. A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinform*. 2006;7(1):56.
32. 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*. Springer; 2017. p. 310–328.
33. Le Novère N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B, et al. *Biomodels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems*. *Nucleic Acids Res*. 2006;34(suppl–1):689–91.
34. Argyris G, Lluch Lafuente A, Tribastone M, Tschaikowski M, Vandin A. Reducing boolean networks with backward Boolean equivalence. In: *International conference on computational methods in systems biology*. Springer; 2021. p. 1–18.
35. Cardelli L, Tribastone M, Tschaikowski M, Vandin A. Symbolic computation of differential equivalences. In: *POPL 2016*. 2016. p. 137–150. <https://doi.org/10.1145/2837614.2837649>.
36. De Moura L, Bjørner N. Z3: an efficient SMT solver. In: *International conference on tools and algorithms for the construction and analysis of systems*. Springer; 2008. p. 337–340.
37. Naldi A, Hernandez C, Levy N, Stoll G, Monteiro PT, 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. *Front Physiol*. 2018;9:680. <https://doi.org/10.3389/fphys.2018.00680>.
38. Naldi A, Monteiro PT, Müssel C, Kestler HA, Thieffry D, Xenarios I, Saez-Rodríguez J, Helikar T, Chaouiya C, for Logical Models C, Tools. Cooperative development of logical modelling standards and tools with CoLoMoTo. *Bioinformatics*. 2015;31(7):1154–9.
39. Klarner H, Streck A, Siebert H. PyBoolNet: a python package for the generation, analysis and visualization of Boolean networks. *Bioinformatics*. 2017;33(5):770–2.
40. Chaouiya C, Naldi A, Thieffry D. Logical modelling of gene regulatory networks with GINsim. 2012;804:463–79.
41. Di Cara A, Garg A, De Micheli G, Xenarios I, Mendoza L. Dynamic simulation of regulatory networks using squad. *BMC Bioinform*. 2007;8(1):462.

42. ...Malik-Sheriff RS, Glont M, Nguyen TVN, Tiwari K, Roberts MG, Xavier A, Vu MT, Men J, Maire M, Kananathan S, Fairbanks EL, Meyer JP, Arankalle C, Varusai TM, Knight-Schrijver V, Li L, Dueñas-Roca C, Dass G, Keating SM, Park YM, Buso N, Rodriguez N, Hucka M, Hermjakob H. BioModels—15 years of sharing computational models in life science. *Nucleic Acids Res.* 2020;48(D1):407–15. <https://doi.org/10.1093/nar/gkz1055.gkz1055>.
43. Fauré A, Vreede B, Sucena E, Chaouiya C. A discrete model of *Drosophila* eggshell patterning reveals cell-autonomous and Juxtacrine effects. *PLoS Comput Biol.* 2014;10:1003527. <https://doi.org/10.1371/journal.pcbi.1003527>.
44. Delaplace F, Ivanov S. Bisimilar Booleanization of multivalued networks. *BioSystems*, 2020;104205
45. 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.* 2013;9(10):1003286.
46. Coolen M, Thieffry D, Drivenes Ø, Becker TS, Bally-Cuif L. miR-9 controls the timing of neurogenesis through the direct inhibition of antagonistic factors. *Dev Cell.* 2012;22(5):1052–64.
47. Wittmann DM, Krumsiek J, Saez-Rodriguez J, Lauffenburger DA, Klamt S, Theis FJ. Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC Syst Biol.* 2009;3(1):98. <https://doi.org/10.1186/1752-0509-3-98>.
48. Sipser M. Introduction to the theory of computation. 3rd ed. Boston: Course Technology; 2013.
49. Müssel C, Hopfensitz M, Kestler HA. BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics.* 2010;26(10):1378–80.
50. Mbodj A, Junion G, Brun C, Furlong EE, Thieffry D. Logical modelling of *Drosophila* signalling pathways. *Mol BioSyst.* 2013;9(9):2248–58.
51. Martinez-Sanchez ME, Hiriart M, Alvarez-Buylla ER. The CD4+T cell regulatory network mediates inflammatory responses during acute hyperinsulinemia: a simulation study. *BMC Syst Biol.* 2017;11(1):1–12.
52. Mbodj A, Gustafson EH, Ciglar L, Junion G, Gonzalez A, Girardot C, Perrin L, Furlong EE, Thieffry D. Qualitative dynamical modelling can formally explain mesoderm specification and predict novel developmental phenotypes. *PLoS Comput Biol.* 2016;12(9):1005073.
53. Martinez-Sanchez ME, Mendoza L, Villarreal C, Alvarez-Buylla ER. A minimal regulatory network of extrinsic and intrinsic factors recovers observed patterns of CD4+ T cell differentiation and plasticity. *PLoS Comput Biol.* 2015;11(6):1004324.
54. Fauré A, Vreede BM, Sucena É, Chaouiya C. A discrete model of *Drosophila* eggshell patterning reveals cell-autonomous and Juxtacrine effects. *PLoS Comput Biol.* 2014;10(3):1003527.
55. Sánchez L, Chaouiya C. Primary sex determination of placental mammals: a modelling study uncovers dynamical developmental constraints in the formation of Sertoli and granulosa cells. *BMC Syst Biol.* 2016;10(1):1–11.
56. Mombach JC, Bugs CA, Chaouiya C. Modelling the onset of senescence at the G1/S cell cycle checkpoint. *BMC Genom.* 2014;15(S7):7.
57. Corral-Jara KF, Chauvin C, Abou-Jaoudé W, Grandclaude M, Naldi A, Soumelis V, Thieffry D. Interplay between smad2 and stat5a is a critical determinant of IL-17A/IL-17F differential expression. *Mol Biomed.* 2021;2(1):1–16.
58. Abou-Jaoudé W, Monteiro PT, Naldi A, Grandclaude M, Soumelis V, Chaouiya C, Thieffry D. Model checking to assess T-helper cell plasticity. *Front Bioeng Biotechnol.* 2015;2:86.
59. Kondratova M, Barillot E, Zinoviyev A, Calzone L. Modelling of immune checkpoint network explains synergistic effects of combined immune checkpoint inhibitor therapy and the impact of cytokines in patient response. *Cancers.* 2020;12(12):3600.
60. Vaga S, Bernardo-Faura M, Cokelaer T, Maiolica A, Barnes CA, Gillet LC, Hegemann B, van Drogen F, Sharifian H, Klipp E, et al. Phosphoproteomic analyses reveal novel cross-modulation mechanisms between two signaling pathways in yeast. *Mol Syst Biol.* 2014;10(12):767.
61. Naldi A, Carneiro J, Chaouiya C, Thieffry D. Diversity and plasticity of TH cell types predicted from regulatory network modelling. *PLoS Comput Biol.* 2010;6(9):1000912.
62. Nuñez-Reza KJ, Naldi A, Sánchez-Jiménez A, Leon-Apodaca AV, Santana MA, Thomas-Chollier M, Thieffry D, Medina-Rivera A. Logical modelling of in vitro differentiation of human monocytes into dendritic cells unravels novel transcriptional regulatory interactions. *Interface focus.* 2021;11(4):20200061.
63. Terfve C, Cokelaer T, Henriques D, MacNamara A, Goncalves E, Morris MK, Iersel MV, Lauffenburger DA, Saez-Rodriguez J. CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst Biol.* 2012;6(1):1–14.
64. Floc'Hlay S, Molina MD, Hernandez C, Haillot E, Thomas-Chollier M, Lepage T, Thieffry D. Deciphering and modelling the TGF- β signalling interplays specifying the dorsal-ventral axis of the sea urchin embryo. *Development.* 2021;148(2): 189944.
65. Hernandez C, Thomas-Chollier M, Naldi A, Thieffry D. Computational verification of large logical models-application to the prediction of t cell response to checkpoint inhibitors. *bioRxiv.* 2020.
66. Fauré A, Naldi A, Lopez F, Chaouiya C, Ciliberto A, Thieffry D. Modular logical modelling of the budding yeast cell cycle. *Mol BioSyst.* 2009;5(12):1787–96.
67. Calzone L, Tournier L, Fourquet S, Thieffry D, Zhivotovsky B, Barillot E, Zinoviyev A. Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS Comput Biol.* 2010;6(3):1000702.
68. Niarakis A, Bounab Y, Grieco L, Roncagalli R, Hesse A-M, Garin J, Malissen B, Daëron M, Thieffry D. Computational modeling of the main signaling pathways involved in mast cell activation. *Fc Recept.* 2014;69–93
69. Sahin Ö, Fröhlich H, Löbke C, Korf U, Burmester S, Majety M, Mattern J, Schupp I, Chaouiya C, Thieffry D, et al. Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC Syst Biol.* 2009;3(1):1.
70. MacNamara A, Terfve C, Henriques D, Bernabé BP, Saez-Rodriguez J. State-time spectrum of signal transduction logic models. *Phys Biol.* 2012;9(4): 045003.
71. Selvaggio G, Canato S, Pawar A, Monteiro PT, Guerreiro PS, Brás MM, Janody F, Chaouiya C. Hybrid epithelial-mesenchymal phenotypes are controlled by microenvironmental factors. *Can Res.* 2020;80(11):2407–20.

72. Chaouiya C, Bérenguier D, Keating SM, Naldi A, Van Iersel MP, 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 Syst Biol.* 2013;7(1):1–15.
73. Remy E, Rebouissou S, Chaouiya C, Zinovyev A, Radvanyi F, Calzone L. A modeling approach to explain mutually exclusive and co-occurring genetic alterations in bladder tumorigenesis. *Can Res.* 2015;75(19):4042–52.
74. González A, Chaouiya C, Thieffry D. Dynamical analysis of the regulatory network defining the dorsal-ventral boundary of the drosophila wing imaginal disc. *Genetics.* 2006;174(3):1625–34.
75. Sánchez L, Chaouiya C. Logical modelling uncovers developmental constraints for primary sex determination of chicken gonads. *J R Soc Interface.* 2018;15(142):20180165.
76. Hamey FK, Nestorowa S, Kinston SJ, Kent DG, Wilson NK, Göttgens B. Reconstructing blood stem cell regulatory network models from single-cell molecular profiles. *Proc Natl Acad Sci.* 2017;114(23):5822–9.
77. Béal J, Pantolini L, Noël V, Barillot E, Calzone L. Personalized logical models to investigate cancer response to BRAF treatments in melanomas and colorectal cancers. *PLoS Comput Biol.* 2021;17(1):1007900.
78. Simao E, Remy E, Thieffry D, Chaouiya C. Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in *E. coli*. *Bioinformatics.* 2005;21(suppl-2):190–6.
79. Enciso J, Mayani H, Mendoza L, Pelayo R. Modeling the pro-inflammatory tumor microenvironment in acute lymphoblastic leukemia predicts a breakdown of hematopoietic-mesenchymal communication networks. *Front Physiol.* 2016;7:349.
80. Sánchez L, Thieffry D. A logical analysis of the drosophila gap-gene system. *J Theor Biol.* 2001;211(2):115–41.
81. Fauré A, Thieffry D. Logical modelling of cell cycle control in eukaryotes: a comparative study. *Mol Biosyst.* 2009;5(12):1569–81.
82. Mendoza L. A network model for the control of the differentiation process in TH cells. *Biosystems.* 2006;84(2):101–14.
83. González A, Chaouiya C, Thieffry D. Logical modelling of the role of the HH pathway in the patterning of the drosophila wing disc. *Bioinformatics.* 2008;24(16):234–40.
84. Sánchez L, Chaouiya C, Thieffry D. Segmenting the fly embryo: logical analysis of the role of the segment polarity cross-regulatory module. *Int J Dev Biol.* 2002;52(8):1059–75.
85. Montagud A, Béal J, Tobalina L, Traynard P, Subramanian V, Szalai B, Alföldi R, Puskás L, Valencia A, Barillot E, Saez-Rodríguez J, Calzone L. Patient-specific Boolean models of signaling networks guide personalized treatments. *bioRxiv.* 2021. <https://doi.org/10.1101/2021.07.28.454126>.
86. Sánchez-Villanueva JA, Rodríguez-Jorge O, Ramírez-Pliego O, Rosas Salgado G, Abou-Jaoudé W, Hernandez C, Naldi A, Thieffry D, Santana MA. Contribution of ROS and metabolic status to neonatal and adult CD8+ T cell activation. *PLoS ONE.* 2019;14(12):0226388.
87. Verlingue L, Dugourd A, Stoll G, Barillot E, Calzone L, Londoño-Vallejo A. A comprehensive approach to the molecular determinants of lifespan using a Boolean model of geroconversion. *Aging Cell.* 2016;15(6):1018–26.
88. Flobak Å, Baudot A, Remy E, Thommesen L, Thieffry D, Kuiper M, Lægreid A. Discovery of drug synergies in gastric cancer cells predicted by logical modeling. *PLoS Comput Biol.* 2015;11(8):1004426.
89. Zañudo JG, Steinway SN, Albert R. Discrete dynamic network modeling of oncogenic signaling: mechanistic insights for personalized treatment of cancer. *Curr Opin in Syst Biol.* 2018;9:1–10.
90. Cohen DP, Martignetti L, Robine S, Barillot E, Zinovyev A, Calzone L. Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Comput Biol.* 2015;11(11):1004571.
91. Cacace E, Collombet S, Thieffry D. Logical modeling of cell fate specification—application to t cell commitment 2020;139:205–238
92. Collombet S, van Oevelen C, Ortega JLS, Abou-Jaoudé W, Di Stefano B, Thomas-Chollier M, Graf T, Thieffry D. Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proc Natl Acad Sci.* 2017;114(23):5792–9.
93. Traynard P, Fauré A, Fages F, Thieffry D. Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation. *Bioinformatics.* 2016;32(17):772–80.
94. Abou-Jaoudé W, Ouattara DA, Kaufman M. From structure to dynamics: frequency tuning in the P53-MDM2 network: I. logical approach. *J Theor Biol.* 2009;258(4):561–77.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

