



**HAL**  
open science

# TaBooN - Boolean Network Synthesis Based on Tabu Search

Sara Sadat Aghamiri, Franck Delaplace

► **To cite this version:**

Sara Sadat Aghamiri, Franck Delaplace. TaBooN - Boolean Network Synthesis Based on Tabu Search. 2020. hal-02933067

**HAL Id: hal-02933067**

**<https://hal.science/hal-02933067>**

Preprint submitted on 8 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# TaBooN

## Boolean Network Synthesis Based on Tabu Search

Sara Sadat Aghamiri<sup>1</sup> and Franck Delaplace <sup>\*2</sup>

<sup>1,2</sup>Paris-Saclay University -Univ. Evry, IBISC laboratory

### Abstract

Recent developments in Omics-technologies revolutionized the investigation of biology by producing molecular data in multiple dimensions and scale. This breakthrough in biology raises the crucial issue of their interpretation based on modelling. In this undertaking, network provides a suitable framework for modelling the interactions between molecules. Basically a Biological network is composed of nodes referring to the components such as genes or proteins, and the edges/arcs formalizing interactions between them. The evolution of the interactions is then modelled by the definition of a dynamical system. Among the different categories of network, the Boolean network offers a reliable qualitative framework for the modelling. Automatically synthesizing a Boolean network from experimental data therefore remains a necessary but challenging issue.

In this study, we present TaBooN, an original work-flow for synthesizing Boolean Networks from biological data. The methodology uses the data in the form of Boolean profiles for inferring all the potential local formula inference. They combine to form the model space from which the most truthful model with regards to biological knowledge and experiments must be found. In the TaBooN work-flow the selection of the fittest model is achieved by a Tabu-search algorithm. TaBooN is an automated method for Boolean Network inference from experimental data that can also assist to evaluate and optimize the dynamic behaviour of the biological networks providing a reliable platform for further modelling and predictions.

**Keywords :** Boolean Network, Model Synthesis, Tabu-Search

---

\*Corresponding author

# 1 Introduction

Boolean networks (BNs) study applied to biology has been pioneered by S. Kauffman [1, 2] and R. Thomas [3, 4] as a regulation network modelling approach. The BNs consist of a logical dynamical system formalizing the interactions between the elements of a regulatory network in terms of discrete variables, logical functions and parameters. Biological studies benefit from such networks to structure and represent the molecular interactions with the synthesis of BNs where the nodes refer to the components such as genes or proteins, and the edges show the interactions between them [5, 6, 7]. Also different expressions and productions of the molecules in various conditions can be characterized as on/active/up-regulated or off/inactive/down-regulated in BNs. BNs have been considered as a reliable standard approach to study signaling and regulatory networks, by modelling and analysing biological processes [8, 9, 10, 11, 12].

In the recent decades, technological advancements in data generation have produced unprecedented accelerating amounts of experimental data [13]. These technologies revolutionized the investigation of biology and human health, producing data in multiple dimensions and scale including DNA sequence, epigenomic states, single-cell gene expression activity, proteomics, functional, and phenotypic measurements [14, 15]. The biological information accumulated over the years in databases and literature as a static source of knowledge [16, 17] that however provides limited insight into the system's response to perturbations [18].

Although the availability of biological information plays a crucial role in the construction of biological networks, the procedure of synthesizing a biological BN manually can take a considerable amount of time and could be error prone. Indeed, in a manual construction of a regulatory network requires an iterated improvement of the network, validated by experimental data and biological knowledge that could reveal daunting due to the successive revisions for fitting with the validation elements. Therefore, automating this procedure based on experimental data for synthesizing a biological BN, is an essential approach [19].

Different approaches have been applied to automate and optimize the construction of these biological BNs. In [20] researchers proposed a pipeline for implementing a Boolean logic model from a "prior knowledge network" (PKN, *ie.*, , a network obtained from literature or expert knowledge) and trains it upon perturbation data. The [20] analysis includes the import of the network and data, processing the network, training, and reporting the analysis results. This method can features different logic formalisms from

Boolean models to differential equations in a common framework.

The approach to generate and optimize Boolean networks, based on a given PKN is also used by [17]. This method utilizes an optimization approach to produce specific, contextualized models from generic PKNs. This procedure includes implementing a genetic algorithm to construct a model network as a sub-network of the PKN and trained against experimental data to reproduce the experimentally observed behaviour in terms of attractors and the transitions due to specific perturbations. The resulting model network forms a dynamic Boolean model that is more similar to the observed biological process used to train the model than the original PKN.

Other methodologies in this field [21] addresses the construction of Boolean functions from constraints on their domain and emerging dynamic properties of the resulting network. The existence and absence of trajectories between partially observed configurations, and stable behaviours (fixpoints and cyclic attractors) are associated with the dynamic properties. The construction of BNs in [21] expressed as a Boolean satisfiability problem relying on Answer-Set Programming with a parametrized complexity leading to a complete non-redundant characterization of the set of solutions.

Applying Answer Set Programming is also used by [22]. Describing that an efficient and scalable training method focuses on the comparison of two time-points and assumes that the system has reached an early steady-state, [22] generalizes such a learning procedure to take into account the time series traces of phosphoproteomics data to discriminate Boolean networks according to their transient dynamics. This identifies a specific condition that must be satisfied by the dynamics of a Boolean network to be compatible with a discretized time series trace. The methodology includes an Answer Set Programming to compute an over-approximation of the Boolean network set that fits best with experimental data and provides the corresponding encodings. This procedure, combined with model-checking approaches, points to a global learning algorithm.

In this field, another approach described by [23] utilizes Satisfiability Modulo Theory. This method defines automated formal reasoning, which permits the construction and analysis of the complete set of logical models consistent with experimental observations. The procedure of this methodology includes identifying critical network components, defining definite and possible interactions, characterizing an Abstract Boolean Network, encoding experimental observations as constraints on state trajectories, enumerating the concrete models that satisfy these constraints, identifying minimal networks with specific features, along with prediction steps. As a result, this methodology transforms knowledge of complex biological processes from sets

of possible interactions and experimental observations to predictive biological programs governing cell function.

In [24], the authors defines a Boolean network inference method from time-series gene expression data using a genetic algorithm, called GABNI. The introduced method exploited an existing method, MIBNI, in the first stage to find an optimal solution and then GABNI if the first method fails due to the degree of complexity of an underlying regulatory function. Also in [25] Barman and his colleagues propose a pipeline for the gene regulatory network inference from time-series gene expression data by applying a statistical method called the chi-square method to infer a Boolean network from time-series gene expression data. They also suggested that structural accuracy can be increased by combining the chi-square test with neural networks as the perspective of their methodology.

Applying different methods to automate this process may circumvent the limitation of a particular method and potentially improve the overall accuracy of the resulting BN [19]. Besides benefiting from Boolean network inference from experimental data can also assist to evaluate and optimize the dynamic behaviour of the biological networks which provides a more reliable platform for further *in silico* experiments and predictions [26].

In this study, we present TaBoon, an original work-flow for synthesizing BNs from biological data. This methodology utilizes biological information or experimental data in the form of the Boolean profiles for optimizing local formula inference and then Tabu-search algorithm to select the best BN candidate regarding significant biological features that have been defined for the system, viewed as global properties that the biological system complied. These properties are usually used after the synthesis for validating the model. In our approach they are employed for the synthesis insuring that the generated model fulfils them by construction.

After recalling the main features of the Boolean network (Section 2) we detail the TaBoon work-flow in Section 3 which contains the description for Binarization 3.1, Local formula inference 3.2 ,and the final synthesis 3.3. In the TaBoon benchmark (Section 4) we present the application of the TaBoon by using biological models befor concluding (Section 5) on the methodology and functionality of TaBoon pipeline for automatic modelling and validating the biological BNs.

## 2 Boolean Network

Boolean network is a discrete dynamical system modelling the gene expression activity by capturing the functional transitions between two basic regulatory status: active or not defined by the Boolean values  $\mathbb{B} = \{1, 0\}$  respectively. Formally, it operates on Boolean variables  $X$  by determining their *state* evolution where a state  $s$  is an interpretation assigning a Boolean value to the variables (*ie.*,  $s : X \rightarrow \mathbb{B}$ ).  $\mathbb{B}_X$  denotes the set of all states for a set of variables  $X$ . A Boolean network is defined by a collection of Boolean functions,  $F = \{x_i = f_i(x_1, \dots, x_n) \mid 1 \leq i \leq n\}$ , where each  $f_i : \mathbb{B}_X \rightarrow \mathbb{B}_{x_i}$  is a propositional formula computing the state of  $x_i$ .

**Model of dynamics.** The model of dynamics describes the state evolutions by a labelled transition system where the states are updated with respect to an updating policy, called the *mode*. For example, in the *asynchronous* mode, a single variable is updated per transition. Hence, the transition system is  $\langle \longrightarrow, X, \mathbb{B}^n \rangle$  where the transition relation  $\longrightarrow \subseteq \mathbb{B}_X \times X \times \mathbb{B}_X$  is labelled by the updated variable such that<sup>1</sup>:  $s \xrightarrow{x_i} s' \stackrel{\text{def}}{=} s' = (f_i(s) \cup s_{-x_i})$ . Then, the global transition relation is defined as:  $\longrightarrow = \bigcup_{x_i \in X} \xrightarrow{x_i}$ . A path<sup>2</sup>  $s \longrightarrow^* s'$  characterizes a trajectory from  $s$  to  $s'$ .

An *equilibrium*  $s$  is a particular state which is indefinitely reached once met *ie.*,  $\forall s' \in \mathbb{B}_X : s \longrightarrow^* s' \implies s' \longrightarrow^* s$ . A *stable state*  $s$  is a peculiar equilibrium satisfying the stability condition:  $F(s) = s$ . We denote by  $\Omega_F$  the set of stable states of  $F$  (*ie.*,  $\Omega_F = \{s \in \mathbb{B}_X \mid F(s) = s\}$ ).

**Interaction graph** The *interaction graph*  $\mathcal{G}_F = \langle X, \longrightarrow \rangle$  of network  $F$  portrays the causal dependencies between the variables represented by the *signed interactions* describing the nature of their regulatory activity. We denote the set of the regulators for target  $x_j$  by  $\star \longrightarrow x_j$ , and conversely  $x_i \longrightarrow \star$  the set of targets of  $x_i$ . An interaction  $x_i \longrightarrow x_j$  exists whenever changing the value of  $x_i$  may lead to a change in the value of  $x_j$ :

$$x_i \longrightarrow x_j \stackrel{\text{def}}{=} \exists s, s' \in \mathbb{B}_X : s_{x_i} \neq s'_{x_i} \wedge s_{-x_i} = s'_{-x_i} \wedge f_j(s) \neq f_j(s'). \quad (1)$$

The *signed interaction graph*  $\langle Y, \longrightarrow, \sigma \rangle$  refines the nature of the interactions by signing the arcs with the function  $\sigma : (\longrightarrow) \rightarrow \{-1, 0, 1\}$  to

<sup>1</sup>The complement of a set  $E$  by a subset  $E' \subseteq E$  is noted  $-E' = E \setminus E'$ .

<sup>2</sup> $\longrightarrow^*$  is the reflexive and transitive closure of the transition.

represent a monotone relation between the source and target variables of the interaction (2); either increasing (label 1, denoted '+'), or decreasing (label -1, denoted '-'), or neither (label 0, denoted '±'), and formally defined as:

$$\begin{aligned}
 x_i \xrightarrow{+} x_j &\stackrel{\text{def}}{=} x_i \rightarrow x_j \wedge \\
 \forall s, s' \in \mathbb{B}_X : s_{x_i} \leq s'_{x_i} \wedge s_{-x_i} = s'_{-x_i} &\implies f_j(s) \leq f_j(s') \\
 x_i \xrightarrow{-} x_j &\stackrel{\text{def}}{=} x_i \rightarrow x_j \wedge \\
 \forall s, s' \in \mathbb{B}_X : s_{x_i} \leq s'_{x_i} \wedge s_{-x_i} = s'_{-x_i} &\implies f_j(s) \geq f_j(s')
 \end{aligned} \tag{2}$$

### 3 Boolean network synthesis

The algorithm infers a BN from binarized profiles of omic data. It is decomposed in two stages: the *local formula inference* and the *global network synthesis*. The former provides the set of formulas which is consistent with the data for each variable independently from the binarized expression profile and the regulatory graph. The latter selects among the resulting formulas the most appropriate ones in regards to some global properties to validate for the BN. These two stages act complementary by first finding all the consistent formulas compatible with the binarized expression profiles and by then generating a model resulting from an assembly of the formulas by selecting one per variable. Figure 1 summarizes the different steps of the work-flow.

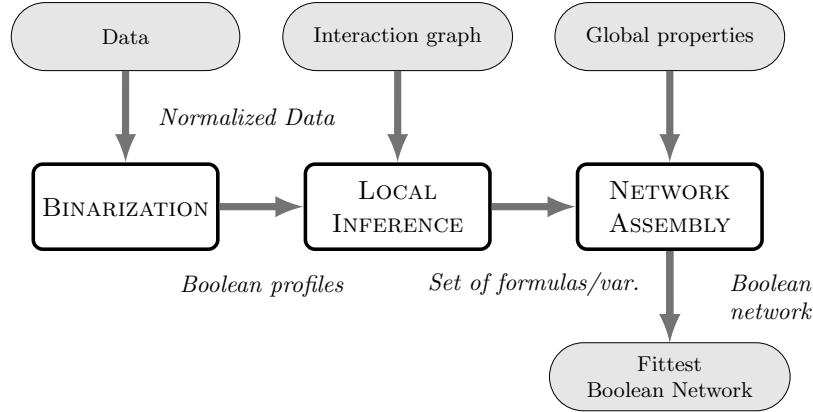


Figure 1: TaBoon work-flow

### 3.1 Binarization

The binarization or Booleanization consists in deducing Boolean profiles from quantified gene expression data. The discretization process can be understood as a partitioning of gene expression value into functional classes respectively representing whether a gene is activated (1) or inhibited (0) that are defined by the value of their expression. Hence the binarization basically corresponds to a bi-partitioning approach where the two classes describe the regulatory status: active or inactive. However, a third class is usually added for collecting gene expression value with status remaining undetermined during analysis. The computational methods differ by the nature of the analysed data: time series [27, 28], differential expression [29, 30], or pseudo-global based on data coming from a set of different conditions [31, 32]. The core of the methods lies on the discovery of a threshold delineating the gene status (active or inactive). We assume that this part is tackled by the above methods and the input of the work-flow is a set of binarized expression profiles.

### 3.2 Local formula inference

For the local inference, the Boolean profiles are assimilated to rows of a Boolean truth table where the profiles of regulators correspond to the input profiles and the regulated variable to the output profiles of the truth table. As example if we do assume that  $x_2, x_3, x_4$  regulate  $x_1$  and we obtain the following profiles  $P = \{(1, 1, 0, 1), (0, 1, 1, 0), (1, 0, -, 0)\}$ , we do consider the following partial truth table (3) for their interpretation:

$x_2$	$x_3$	$x_4$	$x_1$
1	0	1	1
1	1	0	0
0	-	0	1

(3)

Hence, compared to a full truth table, the resulting table is partially defined and possibly with missing values (*ie.*, - sign in Table 3). The local formula inference method will complete the truth table by characterizing all the outputs. This process lies on a set of constraints related to the regulation for discovering all the possible outputs completing the partial truth table from which the formulas consistent with the binarized profiles and the regulation can be deduced. The inference method is thus seen as a satisfiability problem on propositional formula [33] instantiating the output values of the truth table. The issue is thus to define the appropriate set of



constraints for the outputs assimilated to the variables of the satisfiability problem.

By convention, the variables representing the outputs of a truth table are denoted by  $b_p$  and the label  $p$  is the input Boolean profiles. For example given the input profile  $(1, 0, 1)$  the variable representing the output value is denoted  $b_{101}$ .  $x_1$  will always be the output variables while the others ( $x_i, i > 1$ ) are the input variables by convention. Hence, a Boolean profile  $p$  of the regulators yielding to value  $v$  ( $p(x_2, \dots, x_n) = v$ ) for the target  $x_1$  is represented by the equation  $b_p = v$  formulated in propositional logic as an equivalence:  $b_p \iff v$ .

Two kinds of constraints are considered: the definition of the profiles as formulas characterizing the *Boolean profile based constraint*, and the constraints strictly related to the regulation characterizing the *regulation based constraints*.

### Boolean profile based constraint

The Boolean profiles are formulated as logical equivalences (*ie.*,  $b_p \iff v$ ) while however taking the variables with undetermined values into account. We thus need to formalize the equivalences as a combination of defined and undefined values. Recall that when  $v$  is known, the equivalences are simplified according to the following rules  $b \iff 1$  equates  $b$  and  $b \iff 0$  equates  $\neg b$ . Then the equivalence is concretely never explicitly formulated in the constraint related to the profiles while being used for the constraint specification.

Now, assume that the variables with an undetermined value are the variables before rank  $m$ ,  $x_2, \dots, x_m$ , and the values for the other variables are determined,  $x_{m+1}, \dots, x_r$ , we state that at least one variable of the form  $b_{\star, p(x_{m+1}) \dots p(x_r)}$  equals  $p(x_1)$  for a partial Boolean profile  $p$ . This statement is formalized as a disjunction over the valuation of the undefined variables:

$$\mathcal{C}_p^{bp} = \bigvee_{\beta \in \mathbb{B}^{m-1}} (b_{\beta p(x_{m+1}) \dots p(x_r)} \iff p(x_1)). \quad (4)$$

### Regulation based constraints

Two kinds of constraints are deduced from the regulation: the *consensus regulatory profile* and the *regulation conformity* originated from the definition of the interactions.

**Consensus regulatory profile.** This constraint corresponds to a profile where all the regulators cooperate to set the target to a specific value. Two cases are considered:

- all the inhibitors are OFF (0) and all the activators are ON (1), then the target is necessary switched ON according to the definition of the regulation.
- Conversely, when all the inhibitors are ON (1) and all the activators are OFF (0), then target is switched OFF by definition.

We use the labelling convention on  $b_\star$  variables to express them. Assume that the inhibitors correspond to the variables  $R^- = x_2, \dots, x_k$  and  $R^+ = x_{k+1}, \dots, x_r$  are the activators, the constraints are defined by the following formula:

$$\mathcal{C}_{R^-, R^+}^{cp} = \underbrace{b_0 \dots 0}_{k-1} \underbrace{1 \dots 1}_{r-k} \wedge \neg \underbrace{b_1 \dots 1}_{k-1} \underbrace{0 \dots 0}_{r-k} \quad (5)$$

When some regulations remain unknown this constraint is extended similarly to (4) by a disjunction applying all the Boolean configurations for the remaining unknown regulator fulfilling Constraint (5).

**Regulation conformity.** This constraint characterizes the fact that the values of  $b_\star$  variables should also comply to the positive and negative regulation rules (2). This constraint is applied to all regulators independently and expresses the state order characterizing the regulation constraint on the order on labels for  $b_\star$  variables.

Assume that the regulators of  $x_1$  are  $x_2, \dots, x_r$ , the positive interaction  $x_2 \xrightarrow{+} x_1$  leads to the following constraint:

$$\begin{aligned} & \exists s_1, s_2 : \\ & \quad s_1(x_2) < s_2(x_2) \wedge s_{1-x_2} = s_{2-x_2} \wedge f_1(s_1) < f_2(s_2) \quad \wedge \\ & \forall s_1, s_2 : \\ & \quad s_1(x_2) < s_2(x_2) \wedge s_{1-x_2} = s_{2-x_2} \implies f_1(s_1) \leq f_2(s_2). \end{aligned}$$

This condition will be formulated in propositional logic with  $b_\star$  by using the order on the labels of variables. The sole configuration satisfying  $s_1(x_2) < s_2(x_2)$  is  $s_1(x_2) = 0, s_2(x_2) = 1$ . The positive regulation rule must be checked for this configuration only since it holds for the others. By scanning all the remaining configurations stating the equality  $s_{1-x_2} = s_{2-x_2} =$

$\alpha, \alpha \in \mathbb{B}^{r-2}$ , and as  $f_1(s_1) = b_{0\alpha}$  and  $f_1(s_2) = b_{1\alpha}$ , the positive regulation rule is equivalently expressed with  $b_*$  variables as:

$$(\exists \alpha \in \mathbb{B}^{r-2} : b_{0\alpha} < b_{1\alpha}) \wedge (\forall \alpha \in \mathbb{B}^{r-2} : b_{0\alpha} \leq b_{1\alpha}).$$

As the set of regulators is finite, the quantifiers  $\exists, \forall$  can be respectively expressed by  $\bigvee_{\alpha \in \mathbb{B}^{r-2}}, \bigwedge_{\alpha \in \mathbb{B}^{r-2}}$ , and the orders are translated into equivalent formulas:  $\neg x \vee y$  for  $x \leq y$  and  $\neg x \wedge y$  for  $x < y$ . Hence, the condition is formulated in propositional logic as follows:

$$\mathcal{C}_{x_2}^{r+} = \bigvee_{\alpha \in \mathbb{B}^{r-2}} (\neg b_{0\alpha} \wedge b_{1\alpha}) \quad \wedge \quad \bigwedge_{\alpha \in \mathbb{B}^{r-2}} (\neg b_{0\alpha} \vee b_{1\alpha}). \quad (6)$$

Following the same reasoning, the constraint for the negative regulation  $x_2 \xrightarrow{-} x_1$  is:

$$\mathcal{C}_{x_2}^{r-} = \bigvee_{\alpha \in \mathbb{B}^{r-2}} (b_{0\alpha} \wedge \neg b_{1\alpha}) \quad \wedge \quad \bigwedge_{\alpha \in \mathbb{B}^{r-2}} (b_{0\alpha} \vee \neg b_{1\alpha}). \quad (7)$$

### Constraint assembly and formula inference

The final constraint is composed of a conjunction of the previous constraints. Table 1 reports the different constraints used by the inference for the example given introduction of the section (Truth table 3) such that  $x_2$  is a positive regulator,  $x_3$  a negative regulator and the regulation of  $x_4$  is unknown, that is:  $x_2 \xrightarrow{+} x_1, x_3 \xrightarrow{-} x_1, x_4 \xrightarrow{-} x_1$ .

---


$$\begin{aligned} \mathcal{C}_{P(x_1)}^{bp} &= b_{101} \wedge \neg b_{110} \wedge (b_{000} \vee b_{010}). \\ \mathcal{C}_{R^-(x_1), R^+(x_1)}^{cp} &= (b_{100} \wedge \neg b_{010}) \vee (b_{101} \wedge \neg b_{011}). \\ \mathcal{C}_{R^+(x_1)}^{r+} &= \\ &((\neg b_{000} \wedge b_{100}) \vee (\neg b_{001} \wedge b_{101}) \vee (\neg b_{010} \wedge b_{110}) \vee (\neg b_{011} \wedge b_{111})) \wedge \\ &(\neg b_{000} \vee b_{100}) \wedge (\neg b_{001} \vee b_{101}) \wedge (\neg b_{010} \vee b_{110}) \wedge (\neg b_{011} \vee b_{111}) \\ \mathcal{C}_{R^-(x_1)}^{r-} &= \\ &((b_{000} \wedge \neg b_{010}) \vee (b_{001} \wedge \neg b_{011}) \vee (b_{100} \wedge \neg b_{110}) \vee (b_{101} \wedge \neg b_{111})) \wedge \\ &(b_{000} \vee \neg b_{010}) \wedge (b_{001} \vee \neg b_{011}) \wedge (b_{100} \vee \neg b_{110}) \wedge (b_{101} \vee \neg b_{111}) \end{aligned}$$


---

Table 1: Constraints related to the example.

Finally, the resolution of the global constraint is performed by a SAT-solver instantiating all the  $b_*$  variables. Each particular instantiation of the variables thus provides the result of the truth table for each entry. Once the truth table is completed, the minimal disjunctive normal form of a formula is straightforwardly deduced using prime implicant method [34] to generate the resulting formula. The different instances of the  $b_*$  variables define the set of formulas related to the analysed target variable ( $x_1$  here). For the example, three formulas are found (Table 1):

$$\begin{aligned} &(x_2 \wedge x_4) \vee \neg x_3, \\ &(x_2 \wedge \neg x_3) \vee (x_2 \wedge x_4) \vee (\neg x_3 \wedge \neg x_4), \\ &(x_2 \wedge \neg x_3) \vee (\neg x_3 \wedge \neg x_4). \end{aligned}$$

If the regulation of  $x_4$  is known, for example  $x_4 \overset{+}{\dashv} x_1$ , the resolution returns the first formula only.

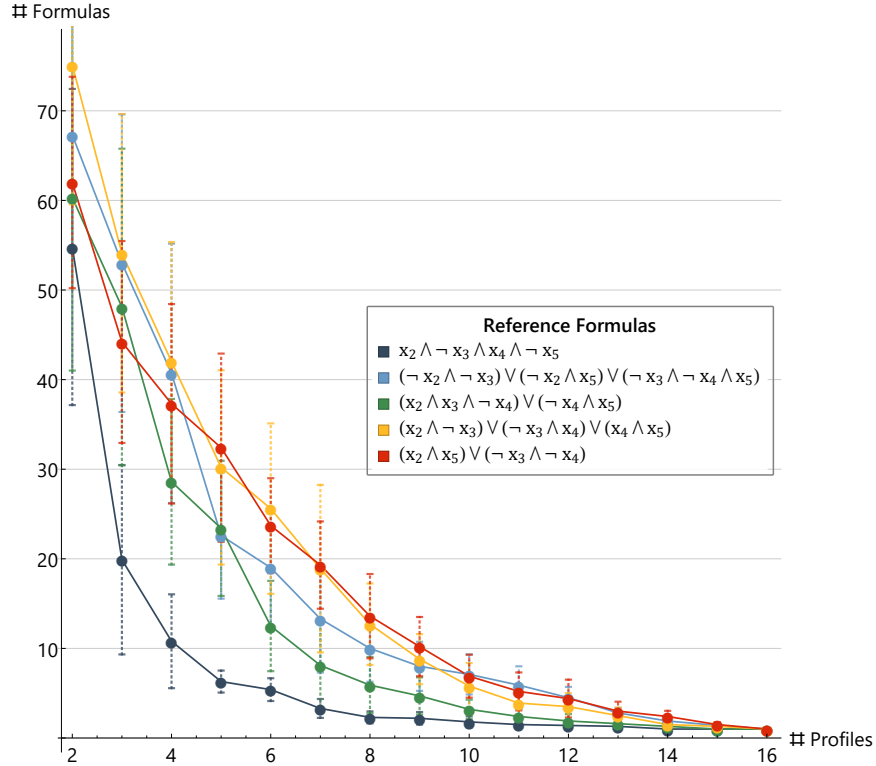
## Experimental evaluation

The accuracy of the method is experimentally evaluated on predefined formulas that the method attempts to discover with Boolean profiles randomly picked from their truth table. Less formulas are inferred more accurate the result is. Figure 2 describes the average number of inferred formulas on 10 trials for 5 different reference formulas<sup>3</sup> involving 4 variables. The number of profiles gradually increases by adding a new profile not already selected to the previous ones at each step.

The method shows a high accuracy in comparison to the number of possible formulas which is  $2^{2^n}$  for  $n$  variables. Indeed, for 4 variables the number of possible formulas is 65 536 and the maximal number of inferred formulas never exceed 100. Thus, the inferred formulas represents less than  $\approx 0.15\%$  of the possible formulas. However, this minute number could remain large as the number of possible formulas is super exponential. The error bars showing the standard deviation reveal a significant sensitivity of the method to the chosen profiles since their number is the same. The means and the standard deviations have an exponential decay stressing the importance of the amount of Boolean profiles provided for the resolution. With a complete set of profile only the tested formula is found. The number of formulas also seemingly depends on the number of terms and clauses. For example, the number of potential formulas for the first reference formula ( $x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg x_5$ ) with the least number of terms is significantly lower

---

<sup>3</sup>Extension of the experiments on a larger set of formulas and trials (not reported here) lead to the same the conclusions.



The curves depict the average number of the inferred formulas from 10 trials of profile sets randomly selected among the lines of the truth table for 5 reference formulas. The error-bars show the standard deviation. All the regulators are defined.

Figure 2: Evaluation of the local inference formula algorithm.

than the others. The complexity of the algorithm is in  $\mathcal{O}(2^{2^n})$  since SAT problem is NP-Complete with  $2^n$  variables representing the outputs of the truth table. However this upper bound does not truly represent the empirical performance of the method which is efficient in practice for  $n \lesssim 10$  variables with a sufficient amount of Boolean profiles. Moreover, the inference can be processed in parallel for a network as the inference is performed independently on each variable.

### 3.3 TaBooN

The final inferred network results from an assembly of the found formulas by selecting one formula per variable. Thus, the number of potential network corresponds to the product of the number of formulas related to variables that may be huge (Figure 4). The selection of a network must therefore be carried out sparingly by selecting the formulas appropriately for yielding a truthful model. This issue is seen as an optimization problem where the putative networks are assessed by quantifying the “truthfulness” with an objective function. TaBooN method uses the Tabu Search [35, 36] that offers a suitable framework for encoding the network inference.

Tabu-search is a meta-heuristic superimposed on another heuristic to explore the best alternative solutions by moving in the neighbourhood of the last found one. Adaptive memory-based strategies are the hallmark of tabu search approaches to circumvent the entrapment in local minima. A memory structure storing the Tabu moves prevents to repeatedly visit the same moves consecutively in order to avoid staying in local minima. Mid and long term memories are respectively used for the intensification around the elite solutions or the diversification opening to alternative solutions.

For the inference of model, the search space represents all the Boolean networks and two Boolean networks are neighbours if and only if they differ in one formula. Therefore a move consists in changing one formula for one variable. To specialize the Tabu search for network inference, we need to define the objective function scoring the truthfulness of a model and the local heuristic search procedure determining the best local move from the current network.

#### Objective function

The objective function quantifies the truthfulness of a network model based on properties used to validate model compared to the feature of the studied biological system or function. Although no specific rules govern the definition of the objective function, some patterns related to the validation principle of a Boolean network drives its definition.

Frequently, a Boolean network is validated by the Boolean signatures defining the molecular states of the phenotypes for some biomarkers [37, 38, 39, 40, 41, 42]. These signatures are expected to be met at stable states in a model because the reproducibility of the phenotypes is modelled by the stability condition, thus assimilating the signatures as a part of the stable states. Notice, that the determination of the stable states can be processed

efficiently by a symbolical computation using SAT solver [43].

Another criterium concerns the monotony of the network. A monotone model contains only pure activations and inhibitions providing a clean model reducing its complexity by limiting the non-monotone interactions.

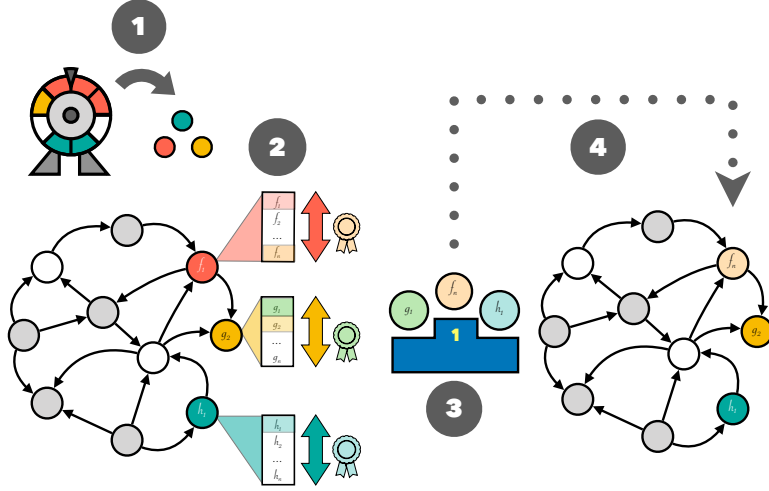
A third criterium assessing the specificity of a model is the number of stable states that should corresponds to the number of the desired phenotypes controlled by the network. Indeed, It is assumed in modelling that each phenotype must correspond to an equilibrium. Hence, models with a large number of stable states are considered as a coarse specification of the biological function because many phenotypic alternatives are carried out by this network. Therefore, the underlying dynamics does not accurately reflect the studied function.

Hence, several criteria are considered for selecting the fittest network leading a multi-criteria function objective. Therefore the comparison between scores is achieved by the minimum of Pareto.

### Move procedure

The local move consists in replacing one formula by another for the current Boolean network. The local search procedure is based on an heuristic selecting the best formula for a network among the sets of formulas computed during the local formula inference phase (Sub-Section 3.2). Due to the potential large number of formulas, we decompose the move in two steps: first some variable are chosen as candidates for the formula change, and then the best formula is elected from the set of formulas of these candidates. The number of candidates at each step is a parameter of the model. Figure 3 summarizes the steps leading to a move.

**Move on candidates** The impact of the modification of the formula differ among candidate depends on the regulation capacity. Therefore, the selection of a candidate is based on a ranking according to their influence in the network. A change of the formula with the highest ranked variable would likely be more consequential on the dynamics than a subsidiary one. Centrality analysis has been widely used to find influential nodes in networks, with a large spectrum of applications in systems biology analysis [44]. The variable ranking is computed from a centrality measure. However, we use the biased roulette wheel method [45] where the size of wheel sections for the variables is proportional to the centrality measurement for privileging the variables with a high centrality while still being able to elect any variable. The probability of choice for a variable with a centrality measure  $c_i$  is thus:



The steps leading to a move are:

1. Select candidates outside the Tabu list (grey nodes) using a biased wheel based on centrality measure.
2. For each candidate select the local best formula by iteratively replacing the initial formula with all the formulas of the candidate and then compare the score of the resulting networks for finding a solution which is Pareto minimal.
3. Elect the move corresponding to the best formula among the local best formulas associated to a candidate.
4. Replace the initial formula by the best formula for the candidate.

Figure 3: Computational steps of a move.

$\frac{c_i}{\sum_{x_j \in X'} c_j}$  where  $X' \subseteq X$  is the set of current available variables outside the tabu list and not already chosen as candidate. We use the Eigenvector centrality measuring the influence of a node in a network for the experiments (Section 4).

**Move on formulas** Once the candidates are selected for a move, the scores are computed by exchanging the formula for each candidate with all their alternative formulas. Then, the best local formula which is a Pareto minimum is chosen. Finally a move corresponds to the selection of the formula with the best score elected from the best local formulas selected for



each candidate. Then, only one formula changes for one candidate during a move.

### **Initialization, diversification and halting condition**

The best network is updated if the score of the current network minimizes it. The halting condition is based on two conditions: A bound on the number of consecutive failures, and reaching the minimal score which is a null vector. The diversification is used for favouring the visit of all the variables. This strategy is motivated by the necessity to adjust all the local dynamics of the variables because they all contribute to get the expected global property. The long term memory stores the frequency of the moves and choose one move which is less frequently visited at each iteration. Finally, the initial network can be designed by a modeller or randomly by selecting one formula per variables. To our knowledge, the quality of the first network speeds the convergence to the best network but does not seem to alter the quality of the final result. Notice that this method may thus be complementary used for improving already designed BNs viewed as the initial network.

## **4 TaBooN benchmark**

The benchmark<sup>4</sup> evaluates the truthfulness of a network obtained by TaBooN with regards to the availability of data (*ie.*, Boolean profiles) and the objective function. To enable a comparative analysis, we evaluate the TaBooN method on five published Boolean networks<sup>5</sup> modelling different biological processes. We consider them as the *references* meaning that they are optimal solutions. Under this hypothesis the truthfulness of a network is assessed by its closeness to a reference network. The main characteristics of the reference networks are summarized in Table 2.

The solution networks are compared to the reference networks by using a distance representing the percent of dissemblances on the result of the evolution functions. The *truth-value distance*  $d_T$  (8) counts the number of differences of the outputs of the truth table for the formulas having the same number of variables with regards to all the possible inputs. Let  $f, f'$  be two formulas with the same number of variables ( $n$ ), the truth-value distance is

---

<sup>4</sup>The parameters used for the experiments are: number of moves= 4, maximal number of formulas = 500, failure bound = 100.

<sup>5</sup>By convention, they will be named by the initial of the three first authors name for their identification.

Name	Modelled biological process	Reference
SFL	ERB EMM receptor regulated G1/S transition network used for anticancer drugs analysis.	[38]
TFF	Network of Mammalian cell cycle	[39]
VDS	Signalling network controlling the S-phase entry and geroconversion senescence.	[40]
CMR	Regulatory network describing Epithelial to Mesenchymal Transition mechanism	[41]
EMM	Network interconnecting the communication pathways between haematopoietic stem cells and mesenchymal stromal cells	[42]

Name of Model	General		Mean degree		Max degree	
	# vertices.	# edges	In	Out	In	Out
SFL	20	51	2.55	2.55	5	6
TFF	11	39	3.55	3.55	6	6
VDS	25	67	2.68	2.68	6	8
CMR	32	157	4.91	4.91	8	13
EMM	26	81	3.12	3.12	9	7

Table 2: Properties of the reference networks.

defined as:

$$d_T(f, f') = \frac{\sum_{s \in \mathbb{B}^n} d_H(f(s), f'(s))}{2^n}; \quad (8)$$

where  $d_H$  is the Hamming distance. For example,  $d_T(x \vee y, x \wedge y) = 0.5$  (50%). Indeed by considering 00, 01, 10, 11 as the sequence of inputs for both formulas the outputs are respectively (0, 1, 1, 1) for OR and (0, 0, 0, 1) for AND. Thus the formulas differ for half of the inputs: 01, 10.

The extension of this distance on networks with the same interaction graph corresponds to the mean of the truth value distances between pairs of formulas defining the evolution of the same variables in each network:

$$d_T(F, F') = \text{mean} \{d_T(f_i, f'_i) \mid f_i \in F, f'_i \in F'\}.$$

The objective function is defined on the three criteria previously mentioned: the occurs of signatures at stable state, the monotony of the model and the number of stable states that should equal the number of stable states

of the reference network ( $\phi$ ). An optimal score corresponds to a null vector.  $\Gamma_B = \{\hat{s}_B \mid \hat{s} \in \Omega_\phi\}$  defines the set of signatures for the biomarkers. Each signature ( $\hat{s}_B$ ) must be included in one stable state for an optimal Boolean network  $F$ .

The objective function for a Boolean network  $F$  returns a vector of scores formally defined in (9) where  $\dot{\cup}$  stands for the concatenation operator. The score of a signature is assessed by the minimum of the Hamming distance ( $d_H$ ) of the stable states to a signature. It is separately applied to each signature providing an independent score for each. Notice that when the signature is included in a stable state, the score is 0 otherwise the function returns the distance of the closest equilibrium to the signature. The monotony is evaluated by counting the number of non-monotone interactions. The score of a monotone network is thus 0. Finally the score related to stable states is the absolute difference between the number of stable states of  $F$  and number of those of the reference network  $\phi$ .

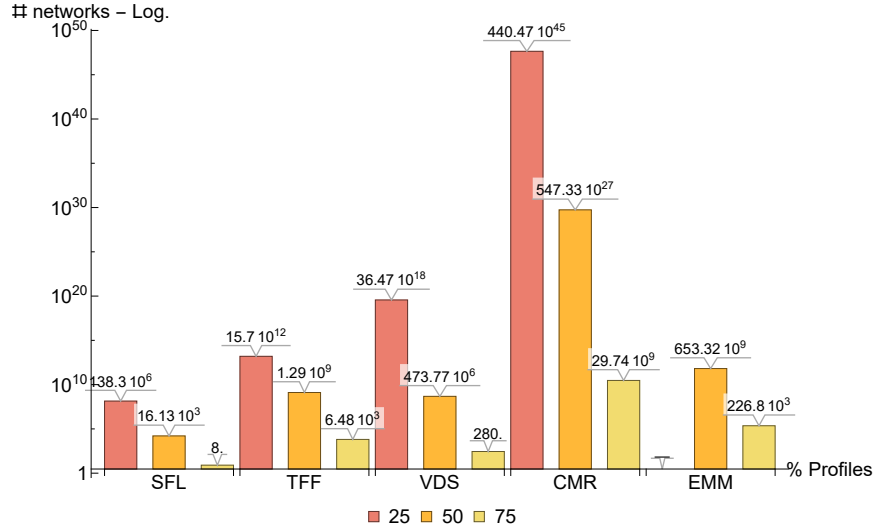
$$\overbrace{\bigcup_{\hat{s}_B \in \Gamma_B} \min\{d_H(\hat{s}_B, s_B) \mid s_B \in \Omega_F\}}^{\text{Biomarker signatures}} \dot{\cup} \overbrace{\sum_{f_i \in F} \mathbf{1}_{\bar{\mathcal{M}}}(f_i)}^{\text{Monotony}} \dot{\cup} \overbrace{\text{abs}(|\Omega_F| - |\Omega_\phi|)}^{\text{Stable states}}, \quad (9)$$

where  $\mathbf{1}_{\bar{\mathcal{M}}}$  is the indicator function of the non-monotone functions (*ie.*,  $\mathbf{1}_{\bar{\mathcal{M}}}(f) = 0$  if  $f$  is monotone and 1 if not). The number of used criteria is thus  $|\Gamma_B| + 2$ .

The goal is to assess the impact of the variation of two parameters: the amount of Boolean profiles representing the binarization of the RNA-seq expression, and the number of biomarkers used for the signatures. We evaluate how the closeness based on the truth-value distance to a reference network defined in Table 2 evolves.

For each network, a given percent of the entries of the truth table is randomly selected and assimilated to the input binary profiles for the formula inference (Sub-Section 3.2). A new set of formulas is computed whenever the profile percent varies. As the biomarkers presumably represent relevant molecules for the study, we order the genes according to a priority stressing their influence determined from their Eigenvector centrality and we incrementally choose a growing number of the biomarkers following this order. The signatures correspond to the values of the chosen biomarkers for each stable state of a reference network. 10 trials are performed for each percent value of binary profiles and signature size. For Boolean profile variation, the size of the biomarkers is fixed to 25% of the number of variables. For

signature variation, the percent of binary profiles is fixed to 50% and the formulas correspond to those computed for this percent.



The bar chart reports the number of potential models (*ie.*, model space cardinality) from which a solution is selected by TaBoon.

Figure 4: Number of potential models.

Figure 5.1 shows the experiments for each trial For 0% of signatures, the formulas are simply randomly selected without computing TaBoon. Figure 5.2 reports the computation time and the number of potential models for each reference network.

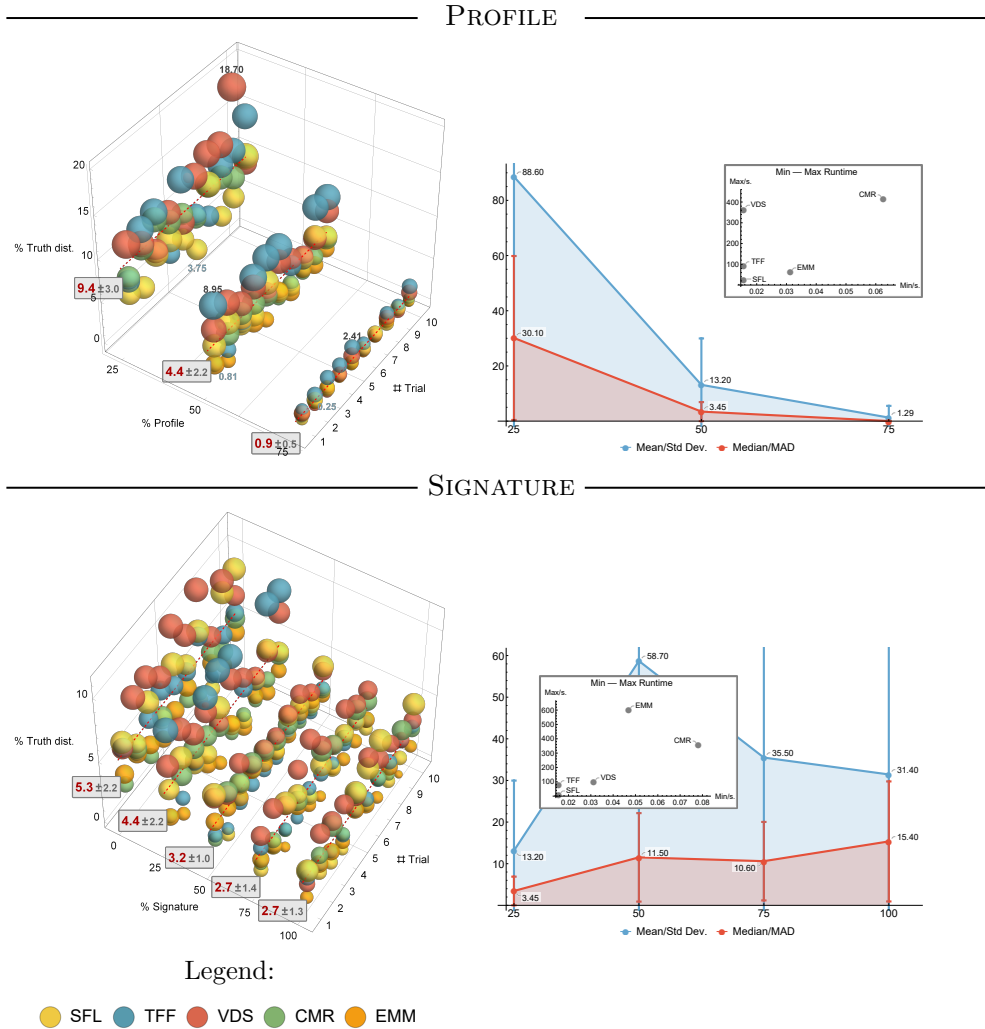
For all trials and all networks, the optimal score is reached (null vector), meaning that no improvement could be achieved for the solution networks and the failure bound was never reached. Therefore the found networks reach the expected signature profiles at equilibria with the same number of stable states as the reference networks. Hence, the solution networks do not differ to reference network for these criteria that are assimilated here the elements validating a model. The truth-value distance remains low with a mean less than 10% at most (25%) and does not exceed 14.20% (Figure 5.1). Compared to the random selection of formulas (Signature = 0%), TaBoon significantly improves the solution since the percent of decrease is respectively 17% for 25% of biomarkers, 40% for 50%, and 49% for 75, 100% with a low initial truth value distance of 5.3% in average. The increase of the number of biomarkers improves the efficiency of the inference. However, we can remark that the score is the same for 75% and 100%.

Finally, the computational time decreases with number of binary profiles (Figure 5.2) since this reduces the number of formulas. However no correlations can be drawn between the time and the size of the signature. The median is low compared to the mean stressing the fact that the computation is efficient with the presence of some outlier executions where reaching the optimal score requires many steps.

TABOON improves the solution compared to a random selection of formulas and the execution time of is reasonable for the tested networks. The obtained truth-value distance indicates that the networks is closed to the dynamics of the reference ones with variation for some entries. This result has to be compared to the number of putative models that corresponds to the product of the number of formulas per nodes (Figure 4). Among a huge number of putative models TABOON is able to find networks having a near-optimal dynamics. Moreover, the major factor of efficiency is related to the amount of data as shown by the evolution of the distance for the binary profiles compared to this of signature (Figure 5.1) and the number of putative models (Figure 4). The increase of the size of the signature also leads to improve the accuracy of the inference until a maximal value that seems to be an asymptote (here 75%) suggesting that the choice of the biokmarkers also matters.

1) *Truth-value distance.*

2) *Computation time.*



1) The experiments are applied on 10 trials for each percent value. In frontside, the mean (in red) and the standard deviation (in gray) are reported for each percent value. In the graphics the minimal (blue) and the maximal (black) distances related to the concerned trials are reported. The diameter of the bubbles indicates the standard deviation ranging from 0.95% (75%) to 19.00% (25%) for the binary profile and from 0.63% (100%) to 15.83% (25%) for the signature with a different scale to the axis for the sake of clarity.

2) The curves show the mean and the median computation time in second of all the networks for the different percentages of profiles and signatures, with the standard deviation and median absolute deviation respectively as error bars. The inserts detail the minimum and maximum duration of each network. The experiments are performed on a quad-core with INTEL I core 7 CPU and 16 Gb of RAM (HP-Zbook 15).

Figure 5: Benchmark results

## 5 Conclusion

Boolean network synthesis is at the core of network based modelling. The synthesis involves a two-stage process: the interaction graph discovery and the dynamical function characterizing the behaviour of the nodes which is a propositional formula for BN. The TaBooN method is focused on the second phases. The characterization of a Boolean network occurs in two stages : the formulas inference that are compatible with the Boolean expression profiles for each node and the election of the fittest formula in each node with regards to global properties related to the molecular dynamics of the studied molecular system. This stage is based on a Tabu meta-heuristic for selecting the best formulas using an objective function for formalizing the global biological properties.

The resulting network gathering all the fittest formulas represents the final outcome of the TaBooN work-flow. As the global properties are based on a model validation protocol, the network with an optimal null score also validates the biological observations otherwise the score estimates a distance to the truthful network. Hence the objective function enables the quantification of the accuracy of a network that can reveal useful for improving the model. A network with a null score thus cannot be contradicted with the validation protocol. Using the classical validation criteria as inputs for the Boolean network synthesis constitutes an originality of the method because they are often used to check the validity of a model without being involved for their synthesis.

The experimental assessment shows the efficiency of the method for finding the most truthful network compared to a random selection of formulas. The benchmark also shows the influence of two parameters: the number of Boolean expression profiles and the precision of the global properties for selecting the best candidate network as final outcome.

It may occur that several networks with an optimal null score can be found. This case can be notably due to the definition of the global properties that insufficiently discriminate the fittest network. Therefore these networks would pave the behavioural space offering alternative models for the analysis. Although only one network should be basically considered as the most truthful, the modelling is thus extended to a pool of networks that cannot be discriminated by the validation process (global properties). Such case addresses as perspective of investigating how to perform relevant modelling with a family of Boolean networks where the remaining uncertainties on the formulas are viewed as unresolved parameters of a network.

## References

- [1] L. Glass and S. A. Kauffman, “The logical analysis of continuous, non-linear biochemical control networks,” *Journal of theoretical Biology*, vol. 39, no. 1, pp. 103–129, 1973.
- [2] S. A. Kauffman, “Metabolic stability and epigenesis in randomly constructed genetic nets,” *Journal of theoretical biology*, vol. 22, no. 3, pp. 437–467, 1969.
- [3] D. Thieffry and R. Thomas, “Dynamical behaviour of biological regulatory networks—ii. immunity control in bacteriophage lambda,” *Bulletin of mathematical biology*, vol. 57, no. 2, pp. 277–297, 1995.
- [4] R. Thomas, D. Thieffry, and M. Kaufman, “Dynamical behaviour of biological regulatory networks—i. biological role of feedback loops and practical use of the concept of the loop-characteristic state,” *Bulletin of mathematical biology*, vol. 57, no. 2, pp. 247–276, 1995.
- [5] A.-L. Barabási, N. Gulbahce, and J. Loscalzo, “Network medicine: a network-based approach to human disease,” *Nature reviews genetics*, vol. 12, no. 1, pp. 56–68, 2011.
- [6] A. Saadatpour and R. Albert, “Boolean modeling of biological regulatory networks: a methodology tutorial,” *Methods*, vol. 62, no. 1, pp. 3–12, 2013.
- [7] I. Albert, J. Thakar, S. Li, R. Zhang, and R. Albert, “Boolean network simulations for life scientists,” *Source code for biology and medicine*, vol. 3, no. 1, pp. 1–8, 2008.
- [8] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry, “Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle,” *Bioinformatics*, vol. 22, no. 14, pp. e124–e131, 2006.
- [9] H. F. Fumia and M. L. Martins, “Boolean network model for cancer pathways: predicting carcinogenesis and targeted therapy outcomes,” *PloS one*, vol. 8, no. 7, p. e69008, 2013.
- [10] R. Schlatter, K. Schmich, I. A. Vizcarra, P. Scheurich, T. Sauter, C. Borner, M. Ederer, I. Merfort, and O. Sawodny, “On/off and beyond—a boolean model of apoptosis,” *PLoS Comput Biol*, vol. 5, no. 12, p. e1000595, 2009.



- [11] M. I. Davidich and S. Bornholdt, “Boolean network model predicts cell cycle sequence of fission yeast,” *PloS one*, vol. 3, no. 2, p. e1672, 2008.
- [12] S. Gupta, S. S. Bisht, R. Kukreti, S. Jain, and S. K. Brahmachari, “Boolean network analysis of a neurotransmitter signaling pathway,” *Journal of theoretical biology*, vol. 244, no. 3, pp. 463–469, 2007.
- [13] R. K. Patel and M. Jain, “Ngs qc toolkit: a toolkit for quality control of next generation sequencing data,” *PloS one*, vol. 7, no. 2, p. e30619, 2012.
- [14] E. R. Mardis, “Next-generation dna sequencing methods,” *Annu. Rev. Genomics Hum. Genet.*, vol. 9, pp. 387–402, 2008.
- [15] Z. Wang, M. Gerstein, and M. Snyder, “Rna-seq: a revolutionary tool for transcriptomics,” *Nature reviews genetics*, vol. 10, no. 1, pp. 57–63, 2009.
- [16] A. Mazein, M. Ostaszewski, I. Kuperstein, S. Watterson, N. Le Novère, D. Lefaudeux, B. De Meulder, J. Pellet, I. Balaur, M. Saqi, *et al.*, “Systems medicine disease maps community driven comprehensive representation of disease mechanisms,” *NPJ systems biology and applications*, vol. 4, no. 1, pp. 1–10, 2018.
- [17] J. Dorier, I. Crespo, A. Niknejad, R. Liechti, M. Ebeling, and I. Xenarios, “Boolean regulatory network reconstruction using literature based knowledge with a genetic algorithm optimization method,” *BMC bioinformatics*, vol. 17, no. 1, p. 410, 2016.
- [18] P. Khatri, M. Sirota, and A. J. Butte, “Ten years of pathway analysis: current approaches and outstanding challenges,” *PLoS Comput Biol*, vol. 8, no. 2, p. e1002375, 2012.
- [19] W.-P. Lee and W.-S. Tzou, “Computational methods for discovering gene networks from expression data,” *Briefings in bioinformatics*, vol. 10, no. 4, pp. 408–423, 2009.
- [20] C. Terfve, T. Cokelaer, D. Henriques, A. MacNamara, E. Goncalves, M. K. Morris, M. van Iersel, D. A. Lauffenburger, and J. Saez-Rodriguez, “Cellnoptr: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms,” *BMC systems biology*, vol. 6, no. 1, pp. 1–14, 2012.

- [21] S. Chevalier, C. Froidevaux, L. Paulevé, and A. Zinovyev, “Synthesis of boolean networks from biological dynamical constraints using answer-set programming,” in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 34–41, IEEE, 2019.
- [22] M. Ostrowski, L. Paulevé, T. Schaub, A. Siegel, and C. Guziolowski, “Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming,” *Biosystems*, vol. 149, pp. 139–153, 2016.
- [23] B. Yordanov, S.-J. Dunn, H. Kugler, A. Smith, G. Martello, and S. Emmott, “A method to identify and analyze biological programs through automated reasoning,” *NPJ systems biology and applications*, vol. 2, no. 1, pp. 1–16, 2016.
- [24] S. Barman and Y.-K. Kwon, “A boolean network inference from time series gene expression data using a genetic algorithm,” *Bioinformatics*, vol. 34, no. 17, pp. i927–1933, 2018.
- [25] S. Barman, H. L. Gope, and Y.-K. Kwon, “A boolean network inference from time-series gene expression data using a statistical method,” in *Proceedings of the International Conference on Computing Advancements*, pp. 1–5, 2020.
- [26] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young, “Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks,” in *Biocomputing 2001*, pp. 422–433, World Scientific, 2000.
- [27] M. Hopfensitz, C. Mussel, C. Wawra, M. Maucher, M. Kuhl, H. Neumann, and H. A. Kestler, “Multiscale binarization of gene expression data for reconstructing boolean networks.,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 9, pp. 487–498, 2012.
- [28] C. Müssel, F. Schmid, T. J. Blätte, M. Hopfensitz, L. Lausser, and H. A. Kestler, “Bitrina–multiscale binarization and trinarization with quality analysis.,” *Bioinformatics (Oxford, England)*, vol. 32, pp. 465–468, Feb. 2016.
- [29] S. Anders and W. Huber, “Differential expression analysis for sequence count data.,” *Genome biology*, vol. 11, p. R106, 2010.

- [30] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, “edger: a bioconductor package for differential expression analysis of digital gene expression data.,” *Bioinformatics (Oxford, England)*, vol. 26, pp. 139–140, Jan. 2010.
- [31] S. Jung, A. Hartmann, and A. del Sol, “RefBool: a reference-based algorithm for discretizing gene expression data,” *Bioinformatics*, vol. 33, pp. 1953–1962, 02 2017.
- [32] J. Béal, A. Montagud, P. Traynard, E. Barillot, and L. Calzone, “Personalization of logical models with multi-omics data allows clinical stratification of patients.,” *Frontiers in physiology*, vol. 9, p. 1965, 2018.
- [33] M. R. Garey and D. S. Johnson, *Computers and intractability*, vol. 29. wh freeman New York, 2002.
- [34] E. J. McCluskey Jr, “Minimization of boolean functions,” *Bell system technical Journal*, vol. 35, no. 6, pp. 1417–1444, 1956.
- [35] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Computers & operations research*, vol. 13, no. 5, pp. 533–549, 1986.
- [36] F. Glover and M. Laguna, “Tabu search,” in *Handbook of combinatorial optimization*, pp. 2093–2229, Springer, 1998.
- [37] L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla, “Genetic control of flower morphogenesis in arabidopsis thaliana: a logical analysis.,” *Bioinformatics (Oxford, England)*, vol. 15, pp. 593–606, 1999.
- [38] Ö. Sahin, H. Fröhlich, C. Löbke, U. Korf, S. Burmester, M. Majety, J. Mattern, I. Schupp, C. Chaouiya, D. Thieffry, *et al.*, “Modeling erbb receptor-regulated gl/s transition to find novel targets for de novo trastuzumab resistance,” *BMC systems biology*, vol. 3, no. 1, p. 1, 2009.
- [39] P. Traynard, A. Fauré, F. Fages, and D. Thieffry, “Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation.,” *Bioinformatics (Oxford, England)*, vol. 32, pp. i772–i780, Sept. 2016.
- [40] L. Verlingue, A. Dugourd, G. Stoll, E. Barillot, L. Calzone, and A. Londoño-Vallejo, “A comprehensive approach to the molecular determinants of lifespan using a boolean model of geroconversion.,” *Aging cell*, vol. 15, pp. 1018–1026, Dec. 2016.

- [41] D. P. A. Cohen, L. Martignetti, S. Robine, E. Barillot, A. Zinovyev, and L. Calzone, “Mathematical modelling of molecular pathways enabling tumour cell invasion and migration.,” *PLoS computational biology*, vol. 11, p. e1004571, Nov. 2015.
- [42] J. Enciso, H. Mayani, L. Mendoza, and R. Pelayo, “Modeling the pro-inflammatory tumor microenvironment in acute lymphoblastic leukemia predicts a breakdown of hematopoietic-mesenchymal communication networks.,” *Frontiers in physiology*, vol. 7, p. 349, 2016.
- [43] E. Dubrova and M. Teslenko, “A sat-based algorithm for finding attractors in synchronous boolean networks,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 8, no. 5, pp. 1393–1399, 2011.
- [44] M. Ashtiani, A. Salehzadeh-Yazdi, Z. Razaghi-Moghadam, H. Hennig, O. Wolkenhauer, M. Mirzaie, and M. Jafari, “A systematic survey of centrality measures for protein-protein interaction networks,” *BMC systems biology*, vol. 12, no. 1, p. 80, 2018.
- [45] M. Srinivas and L. M. Patnaik, “Genetic algorithms: A survey,” *computer*, vol. 27, no. 6, pp. 17–26, 1994.