Analysis of Gene Interactions using Restricted Boolean Networks and Time-Series Data

Carlos H. A. Higa, Vitor H. P. Louzada, and Ronaldo F. Hashimoto

University of São Paulo, São Paulo, SP, Brazil
{higa,louzada,ronaldo}@ime.usp.br

Abstract. A popular model for gene regulatory networks is the Boolean network model. In this paper, we propose an algorithm to perform an analysis of gene regulatory interactions using the Boolean network model and time-series data. Actually, the Boolean network is restricted in the sense that only a subset of all possible Boolean functions are considered. We explore some mathematical properties of the restricted Boolean networks in order to avoid the full search approach. We applied the proposed algorithm in a case study of the budding yeast cell cycle network using an artificial dataset. The results show that some interactions can be fully or, at least, partially determined under the Boolean model considered. We have shown that this analysis can be used as the first step for gene relationships detection with a high flexibility to include biological knowledge. What we envisage with our method is a model that points out which connections should be checked in the wet lab and consequently facilitate some biological experiments.

1 Introduction

Some of the goals of Systems Biology is to study the various cellular mechanisms and components. In many cases, these mechanisms are complex, where some of the interactions between the proteins are still unknown. To represent these interactions it is common to use gene regulatory networks (GRN). There are several models of GRN, from discrete to continuous models. The simplest discrete model was introduced by Kauffman [1] and its known as *Boolean network* model. Later, this model was modified to express uncertainty giving rise to the *probabilistic Boolean network* model [2, 3]. Friedman introduced *Bayesian networks* [4] as a probabilistic tool for the identification of regulatory data and showed that they can reproduce certain known regulatory relationships. Among the continuous models we can cite the *ordinary differential equations* model which was suggested several decades ago [5]. For a more detailed review about models of gene regulatory networks see [6].

Models of gene regulatory networks help us to study biological phenomena (e.g. cell cycle) and diseases (e.g. cancer). Therefore, unreaveling such networks, or at least some of its connections, is an important problem to address. The ability to uncover the mechanisms of GRN has been possible due to developments in high-throughput technologies, allowing scientists to perform analysis

on the DNA and RNA levels. The most common type of data generated by these technologies are gene expression data (microarray).

The biological systems are notoriously complex. Determining how the pieces of this puzzle come together to create living systems is a hard challenge known as *reverse engineering*, which is the process of elucidating the structure of a system by reasoning backwards from observations of its behavior [7]. GRN in many cases cannot be unraveled precisely, however, because of measurement noise and the limited number of data sets compared with the number of genes that are involved.

The most common approach to reverse engineering GRN is to use gene expression data. Some algorithms use additional information from heterogeneous data sources, e.g. genome sequence and protein-DNA interaction data, to assist the inference process. Hecker et al. [8] presents a good review about GRN inference and data integration.

Usually, an inference algorithm aims to construct one single network which is believed to be the real network. The issue is that the inverse problem is ill-posed, meaning that several networks could explain (or generate) the data set given as the input for the algorithm. The problem becomes more complicated if we take into account the noise that may be present in the data and the small amount of samples. For this reason, our approach aims to analyze the network in a statistical manner. Our algorithm creates several networks that could explain the data. By analyzing the similarities among these networks, we will propose a confidence measure of the regulatory relationship between the genes.

In this paper, we present an algorithm based on Boolean networks and time-series gene expression. Actually, the Boolean networks are called *restricted* in the sense that not all Boolean functions are allowed in the model. Restricting the network reduces the search space, which can be significant given that the inverse problem is very complex. The time-series data allow us to observe part of the dynamics of the system. These observations are used to infer the regulatory relationships between the genes.

A challenge always presented in any gene regulatory model is its usefulness. It would be interesting if a model could help biological experiments in understanding gene interactions. The model here presented is capable of inferring some of these connections from time-series data of gene expressions, and this inference process is helped by all *a priori* knowledge available. What we envisage with our method is a model that points out which connections should be determined in the wet lab that would constrain as many other connections as possible and consequently could facilitate some biological experiments.

The paper is organized as follows. In the next section we present the restricted Boolean network model. The algorithm for the statistical analysis is presented in Sect. 3. A budding yeast cell-cycle model from which the artificial data are obtained is described in Sect. 4. In Sect. 5 and 6 we show and discuss our results and we conclude the work in Sect. 7.

2 Restricted Boolean Network Model

A Boolean network (BN) is defined by a set $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ of n Boolean variables and a set $\mathbf{F} = \{f_1, f_2, \ldots, f_n\}$ of n Boolean functions. In the case of GRN the variables are called genes. Obviously, each gene x_i, $i = 1, \ldots, n$, can assume only two possible values: 0 (OFF) or 1 (ON). The value of the gene x_i at time $t+1$ is determined by genes $x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)}$ at time t through a Boolean function $f_i : \{0,1\}^{k_i} \to \{0,1\}$. Given that, there are k_i genes assigned to gene x_i, and the mapping $j_k : \{1, \ldots, n\} \to \{1, \ldots, n\}$, $k = 1, \ldots, k_i$ determines the "wiring" of x_i [9]. This way,

$$x_i(t+1) = f_i(x_{j_1(i)}(t), x_{j_2(i)}(t), \ldots, x_{j_{k_i}(i)}(t)) \ . \tag{1}$$

We assume that all genes are updated synchronously by the functions in $\mathbf{F}$ assigned to them and this process is repeated. The artificial synchrony simplifies computation while preserving the qualitative, generic properties of global network dynamics [11, 10]. A *state* of the network at time t is a binary vector $s(t) = (x_1(t), \ldots, x_n(t))$. Therefore, the number of states is 2^n, labeled by $s_0, s_1, \ldots, s_{2^n-1}$. The dynamics of the network is represented by the transition between states. This model is deterministic given that there is a single Boolean function to regulate each gene. Because of the finite number of states and the deterministic behavior, some of the states may be visited cyclically. These states form what is known by the *attractor* of the BN. The states outside the attractor are called *transient* states. The transient states together with the corresponding attractor states forms the *basin of attraction* of that attractor.

In the case of restricted Boolean networks, the regulatory relationships is represented by a matrix $A_{n\times n}$ using the following convention: $a_{ij} = 1$ for a positive regulation from gene x_j to gene x_i; $a_{ij} = -1$ for a negative regulation from x_j to x_i; For the remaining cases $a_{ij} = 0$. The Boolean function f_i is defined according to the matrix A and the values of the genes x_j, $j = 1, \ldots, n$, at time t:

$$x_i(t+1) = \begin{cases} 1, & \text{if } \sum_j a_{ij} x_j(t) > 0 \\ 0, & \text{if } \sum_j a_{ij} x_j(t) < 0 \\ x_i(t), & \text{if } \sum_j a_{ij} x_j(t) = 0 \ . \end{cases} \tag{2}$$

We call the summation $\sum_j a_{ij} x_j(t)$ the *input* of x_i at time t. Besides the regulatory relationships of the matrix A, each gene can have a self-degradation behavior. A gene x_i with self-degradation is set to 0 whenever its input is null. Observe that not all Boolean functions can be represented using (2) and that is why the Boolean network is called "restricted". In Sect. 4, we will present a budding yeast cell-cycle model proposed by Li et al. [14] which is based on restricted Boolean networks. This model will be used to perform the statistical analysis algorithm.

Example: Let us show a small example of a restricted Boolean network containing only four genes. Fig. 1 shows the regulatory relationship between the four genes. An arrow is a positive regulation; a line with a bar at the end is a negative regulation; the dotted loop on x_2 indicates that this gene has a self-degradation behavior.

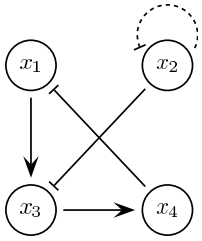

Fig. 1: Small example containing four genes.

Given the regulatory relationships in Fig. 1, the corresponding regulation matrix is presented below:

$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} . \tag{3}$$

Applying the Boolean function given by (2) for every possible state, we can construct a state transition diagram, shown in Fig. 2. As we can see, there are three attractors: 0000, 0001 and 0011; the remaining states are transient states. The attractor 0011 has the largest basin of attraction (we consider the number of states as the size of the basin of attraction).

3 Gene Interaction Analysis Algorithm

The algorithm was designed under the assumption that the gene expression data were generated by a biological system which can be modeled as a restricted Boolean network. Let $\mathbf{S} = \{S(1), S(2), \ldots, S(m)\}$ be a set of m time-series gene expression profiles, where $S(i) \in \{0,1\}^n$ for $i = 1, \ldots, m$. The algorithm aims to analyze networks that produce the sequence

$$S(1) \rightarrow S(2) \rightarrow \cdots \rightarrow S(m) \ . \tag{4}$$

When the network produces the time-series data we say that the network is *consistent* with the data. Naturally, there may exist several consistent networks

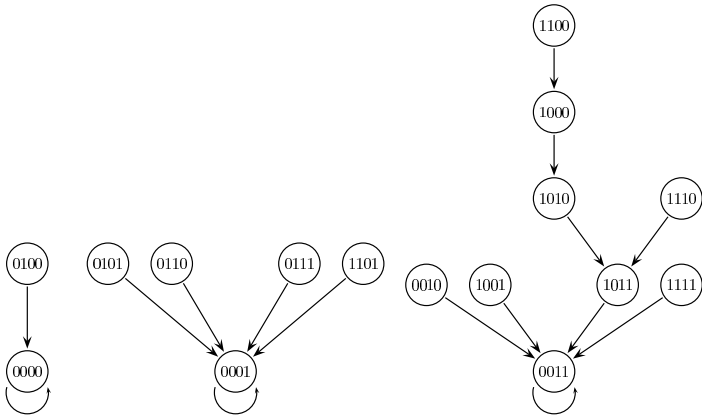

Fig. 2: State transition diagram of the restricted Boolean network shown in Fig. 1.

for a single sequence. That is, the inverse problem is a "one-to-many" or ill-posed problem, and this is very difficult to handle.

One naïve way to solve this ill-posed problem is to find all possible networks by a full search algorithm. In fact, Lau et al. [12] proposed a "smart" full search algorithm to enumerate all possible networks. Here, in this paper, we explore some mathematical properties of the restricted Boolean networks in order to avoid this full search approach.

The algorithm uses an encoding to represent the interaction between a pair of genes. Table 1 shows the code and its respective subset of possible interactions where -1, 0 and 1 stand for *inhibition*, *no relationship* and *activation*, respectively. At the beginning of the process, the relationships between genes are unknown and they are represented by a matrix $A_{n\times n}$ filled with the code 5. This means that any edge (activation or repression) or none can occur (the regulatory relationship is *undetermined*). As the process runs, the entries of the matrix can change to -2, 2 or 3 (*partially determined* relation). In addition, if an entry of the matrix is *completely determined* we can set its value to -1, 0 or 1. At the end of the process the entries of A can hold undetermined, partially determined or determined values. The undetermined and partially determined entries can lead to several matrices that represent a consistent network.

3.1 The Three Steps of the Algorithm

The algorithm aims to uncover the hidden relationships between the genes through the information provided by the time-series sequence, which can be seen as a state transition sequence of the corresponding BN. The algorithm consists in three main steps applied cyclically. Next, we will explain the concepts used in each step.

Table 1: Encoding table used in the algorithm.

Code	Subset
-1	$\{-1\}$
0	$\{0\}$
1	$\{1\}$
-2	$\{-1, 0\}$
2	$\{0, 1\}$
3	$\{-1, 1\}$
5	$\{-1, 0, 1\}$

Step one The first step of the algorithm analyzes the sample in triplets, $S(t-1)$, $S(t)$ and $S(t+1)$. An important point to notice here is that if two consecutive states $S(t-1)$ and $S(t)$ differ only in one single gene x_k, then any gene x_i that had its value changed from $S(t)$ to $S(t+1)$ is directly regulated by x_k. To illustrate this situation, consider the time-series data (Table 2) extracted from the example given in Sect. 2. Looking at the time points $S(1)$ and $S(2)$ we observe that only x_2 had its value changed (from 1 to 0). Now, looking at $S(2)$ and $S(3)$ we can see that x_3 was turned to 1. Following the restricted Boolean network model, this change was caused, necessarily, by the gene x_2. In fact, x_2 inhibits x_3 at time $t = 1$ and it is self degraded at time $t = 2$, allowing x_1 to activate x_3 at time $t = 3$. Using this approach, we state the following proposition (Proposition 1).

Table 2: Time-series data taken from Fig. 2.

t	$x_1(t)$	$x_2(t)$	$x_3(t)$	$x_4(t)$
1	1	1	0	0
2	1	0	0	0
3	1	0	1	0
4	1	0	1	1
5	0	0	1	1

Proposition 1 *Let $S(t-1)$, $S(t)$ and $S(t+1)$ be three consecutive states according to the restricted Boolean network model. If $S(t-1)$ and $S(t)$ differ by a single gene x_k, then for each gene x_i such that $x_i(t) \neq x_i(t+1)$ we have that x_k regulates x_i directly, that is, $a_{ik} \neq 0$.*

Proof. Suppose that $S(t-1)$ and $S(t)$ differ by a single gene x_k, and that there is at least one gene x_i such that $x_i(t) \neq x_i(t+1)$. As $x_i(t) \neq x_i(t+1)$, the summations $\sum_j a_{ij}x_j(t-1)$ and $\sum_j a_{ij}x_j(t)$ have different signs. Given that x_k is the only gene possessing different values in $S(t-1)$ and $S(t)$, this difference signal must have been caused by x_k. Therefore, $a_{ik} \neq 0$.

The type of the regulatory relationship (activation or inhibition) uncovered using Proposition 1 depends on the values of x_k and x_i. Table 3 lists all possible combinations of values for x_k (time $t-1$ and t) and x_i (time t and $t+1$). We call these relationships as *required*, since they must be present in the network in order to maintain the consistency with the time-series data.

Table 3: All possible combinations of values for x_k and x_i.

$x_k(t-1)$	$x_k(t)$	$x_i(t)$	$x_i(t+1)$	type
0	1	0	1	activation
0	1	1	0	inhibition
1	0	0	1	inhibition
1	0	1	0	activation

The approach used in Proposition 1 can be extended when $S(t-1)$ and $S(t)$ differ in more than one gene. For example, let us say that two genes, x_{k_1} and x_{k_2}, are the genes differently expressed from $S(t-1)$ to $S(t)$. If x_i had its value changed from $S(t)$ to $S(t+1)$ there are some *regulation hypotheses* that we must take into account. Analyzing x_{k_1} and x_{k_2} individually, we can use the Table 3 to generate two hypotheses and, these two hypotheses must be combined to generate a third hypothesis. For instance, we can infer that x_{k_1} activates x_i and x_{k_2} inhibits x_i, not in the same network. Given that, a third network would consider both hypotheses simultaneously. This way, the number of hypotheses grows in a combinatorial manner.

Step Two The second step of the algorithm takes into account two consecutive states, $S(t)$ and $S(t+1)$. There is one important observation here: only the active genes at time t can possibly regulate genes at time $t+1$. This fact becomes clear when we look at (2). The active genes can give us an insight of which genes are regulating other gene, although the type of the regulatory relationship can not be determined. However, the input given by the summation in (2) can help us to determine the regulatory relationships. For example, if we observe that a gene x_i changes its value from 0 (at time t) to 1 (at time $t+1$), we can deduce that the input for gene x_i is positive at time t and only the active genes at time t are responsible for this positive input. Following this logic, the algorithm generates all possible combinations of regulatory relationships using the active genes such that the input of gene x_i at time t is coherent to the values of x_i at time $t+1$.

To exemplify, consider the data in Table 2 where $t=3$. At this time, there are two active genes, x_1 and x_3. These genes are the only ones that can contribute to the sign of the input for each gene. If we look at the gene x_4 we observe that its value turned from 0 to 1. According to (2), the input must be positive in this case, that is, $\sum_{j=1}^{4} a_{4j}x_j(3) > 0$. Given that, we must have $a_{41} + a_{43} > 0$. Therefore, neither a_{41} or a_{43} can take the value -1, only 1 or 0 (not both). The

same logic can be applied to all genes and then, the information extracted using this approach can support the inference procedure.

Step Three The third step analyzes any two pairs of consecutive states in the time-series data. Let t_1 and t_2 be two time points in the time-series data:

$$S(1) \rightarrow \cdots \rightarrow S(t_1) \rightarrow S(t_1+1) \rightarrow \cdots \rightarrow S(t_2) \rightarrow S(t_2+1) \rightarrow \cdots \rightarrow S(m) \ . \tag{5}$$

Now, let us suppose that $S(t_1)$ and $S(t_2)$ are very similar. Hence, the difference between $S(t_1+1)$ and $S(t_2+1)$ must be caused by the differentially expressed genes of their predecessors. For instance, let us suppose that $S(t_1)$ and $S(t_2)$ differ in one single gene:

$$S(t_1) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \ , \ S(t_2) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \ . \tag{6}$$

And the succession occurs as stated:

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \ , \cdots \ , \ \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \ . \tag{7}$$

Therefore, the huge difference between $S(t_1+1)$ and $S(t_2+1)$ in this case must be caused by the change on x_4. In this step, the algorithm checks how each gene changed in the two pairs of consecutive states.

In our example, let us concentrate on gene x_1. It was inhibited in the first pair and had no change in the second pair. Let I be the total input originated in the genes with similar expression in $S(t_1)$ and $S(t_2)$, M be the input generated by x_4 in $S(t_1)$, and $\bar{M}$ be the input generated by x_4 in $S(t_2)$. Therefore, to explain the changes of x_1 in the two pairs, we must have:

$$\begin{cases} I + M < 0 \text{ and} \\ I + \bar{M} \geq 0 \ . \end{cases} \tag{8}$$

If a_{ij} represents the influence of gene x_j over x_i, we can calculate I, M and $\bar{M}$ as follows:

$$I = \begin{pmatrix} a_{11} \ a_{12} \ a_{13} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = a_{11} + a_{13} \ , \tag{9}$$

$$M = a_{14} \cdot 0 = 0 \text{ and} \tag{10}$$

$$\bar{M} = a_{14} \cdot 1 = a_{14} \ . \tag{11}$$

Henceforth,

$$\begin{cases} I + M < 0 \\ I + \bar{M} \geq 0 \end{cases} \Longrightarrow \begin{cases} a_{11} + a_{13} + \ 0 \ < 0 \\ a_{11} + a_{13} + a_{14} \geq 0 \end{cases} \Longrightarrow \Big\{ a_{14} > 0 \ . \tag{12}$$

This result implies that the entry a_{14} of the matrix must have the code 1.

If $S(t_1)$ and $S(t_2)$ differ in more than one gene, we can still generate hypotheses of regulation. In fact, this step tries to construct a system of inequalities with the inputs of each gene for every combination of two consecutive pairs.

3.2 Analysis of Gene Interactions

The three steps of the algorithm are performed cyclically until no additional information can be included in the matrix A. At this point, the entries of A are filled with the regulatory hypotheses generated by the algorithm. Some of the entries represent the undetermined or partially determined relationships between genes.

We can think of A as a root of a tree where the leaves are the matrices that can be generated from the root by determining a value for each partially determined/undetermined entry. Perhaps, this value determination can be guided by biological knowledge. In Fig. 3 we show an example using four genes as presented in Sect. 2. There are two partially determined entries in the root (marked with bold face numbers) that can be determined one at a time, generating four possible matrices in the second level of the tree. After determining an entry, the three steps of the algorithm are performed again as previously and the overall process is repeated until a completed determined matrix (a leaf of the tree) is obtained. Some of the leaves are consistent matrices, that is, they represent a network consistent with the data.

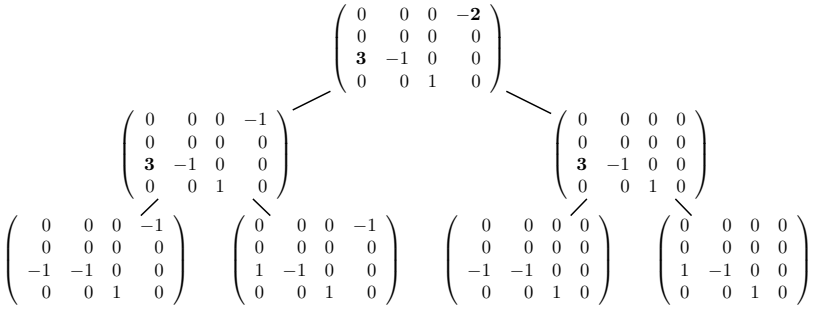

Fig. 3: The root of the tree and the possible matrices generated from the root.

In order to analyze the gene interactions, since there may be a combinatorial explosion in generating the matrices, we randomly generate some of them (a

sampling process) and consider the consistent ones to perform the analysis. In the worst case, this algorithm has exponential running time. However, it does not generates all the 3^{n^2} possible matrices. In the next section, we present a Boolean model of the budding yeast cell cycle that was used to generate artificial data to apply the algorithm and, in Sect. 5, we show the results.

4 Budding Yeast Cell Cycle Model

The cell-cycle process consists of four phases: G_1 (in which the cell grows and, under appropriate conditions, commits to division), S (in which the DNA is synthesized and chromosomes replicated), G_2 (a"gap" between S and M), and M (in which chromosomes are separated and the cell is divided in two). After the M phase, the cell returns to the G_1 phase, waiting for appropriate conditions for another round of division. We call this G_1 phase as stationary G_1. There are ≈ 800 genes involved in the cell-cycle process of the budding yeast [13]. However, the number of key regulators that are responsible for the control and regulation of this complex process is much smaller [14].

The budding yeast cell-cycle model proposed by Li et al. [14] is based on a network of eleven regulators, as shown in Fig. 4. The meaning of the edges are the same as in Fig. 1. The eleven genes $x_1, \ldots, x_{11}$ are Cln3, MBF, SBF, Cln1, Cdh1, Swi5, Cdc20, Clb5, Sic1, Clb1, and Mcm1, respectively. The "cell-size" node was introduced just to indicate a checkpoint to start the cell-cycle process.

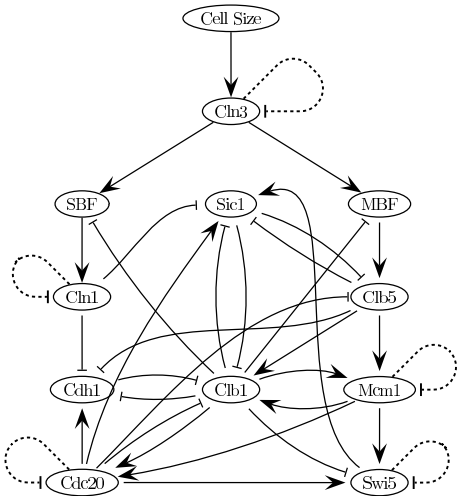

Fig. 4: The cell cycle network of the budding yeast.

Considering the restricted Boolean network model presented in Sect. 2, Li et al. [14] studied the dynamics of the network. They found that there are seven

attractors, shown in Table 4. In this table, each row represents an attractor where the first column indicates the size of the basin of attraction. There is one big basin composed by 1,764 or $\approx 86\%$ of states. According to Li et al. [14], the corresponding attractor is the biological G_1 stationary state.

Table 4: The seven attractors of the cell-cycle network.

Basin size	Cln3	MBF	SBF	Cln1	Cdh1	Swi5	Cdc20	Clb5	Sic1	Clb1	Mcm1
1,764	0	0	0	0	1	0	0	0	1	0	0
151	0	0	1	1	0	0	0	0	0	0	0
109	0	1	0	0	1	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0	1	0	0
7	0	1	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0

Biologically, the cell-cycle sequence starts when the cell commits to division by activating Cln3. To simulate the cell cycle, they started the process by "exciting" the G_1 stationary state with the cell size signal, that is, inducing the gene Cln3 to an active state. Applying (2) to simulate the process it was observed that the system goes back to the G_1 stationary state. The temporal evolution of the states, presented in Table 5, follows the cell-cycle sequence, going from excited G_1 state (Start) to the S phase, the G_2 phase, the M phase, and finally to the stationary G_1 state. This is the biological trajectory or pathway of the cell-cycle network.

Table 5: Temporal evolution of states for the cell-cycle network.

Time	Cln3	MBF	SBF	Cln1	Cdh1	Swi5	Cdc20	Clb5	Sic1	Clb1	Mcm1	Phase
1	1	0	0	0	1	0	0	0	1	0	0	Start
2	0	1	1	0	1	0	0	0	1	0	0	G_1
3	0	1	1	1	1	0	0	0	1	0	0	G_1
4	0	1	1	1	0	0	0	0	0	0	0	G_1
5	0	1	1	1	0	0	0	1	0	0	0	S
6	0	1	1	1	0	0	0	1	0	1	1	G_2
7	0	0	0	1	0	0	1	1	0	1	1	M
8	0	0	0	0	0	1	1	0	0	1	1	M
9	0	0	0	0	0	1	1	0	1	1	1	M
10	0	0	0	0	0	1	1	0	1	0	1	M
11	0	0	0	0	1	1	1	0	1	0	0	M
12	0	0	0	0	1	1	0	0	1	0	0	G_1
13	0	0	0	0	1	0	0	0	1	0	0	Stationary G_1

The states presented in Table 5 are used as the time-series data to perform the statistical analysis. The results are shown in the next section.

5 Results

The application of the algorithm presented in Sect. 3 creates a collection of consistent networks totally inferred from the time-series data of the yeast cell cycle.

If we calculate the frequency of the connections, we are capable of assigning probabilities to each gene relationship. In Fig. 5 and 6 we show the frequency of different types of inward connections to each gene from all other genes. Evidently, the determined connections will appear with frequency 100% in all the networks; while the partially determined connections will have, at least, one gene relationship (activation, no connection or inhibition) with frequency 0%.

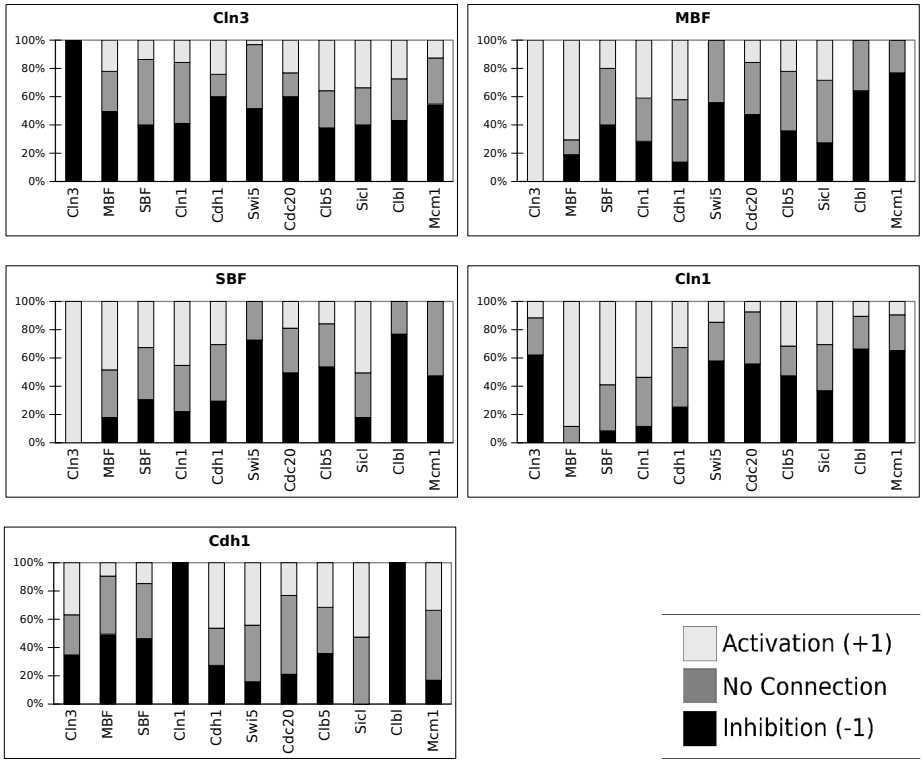

Fig. 5: Frequency of the relationships in the consistent networks. The statistics of inward connections to each gene from all other genes were created by the consecutively application of the three steps of the described algorithm and by a random determination of one connection. The determined connections exhibits only one color (black, white or gray), and the partially determined connections exhibit two colors. 100 networks were used for the statistical analysis. The results for the remaining genes are shown in Fig. 6.

From the frequencies shown in Fig. 5 and 6, we can see that the algorithm was capable of identifying 11 determined connections and 13 partially determined connections. The results are shown in Fig. 7. Note that, in this figure, the arrows do not indicate activation necessarily.

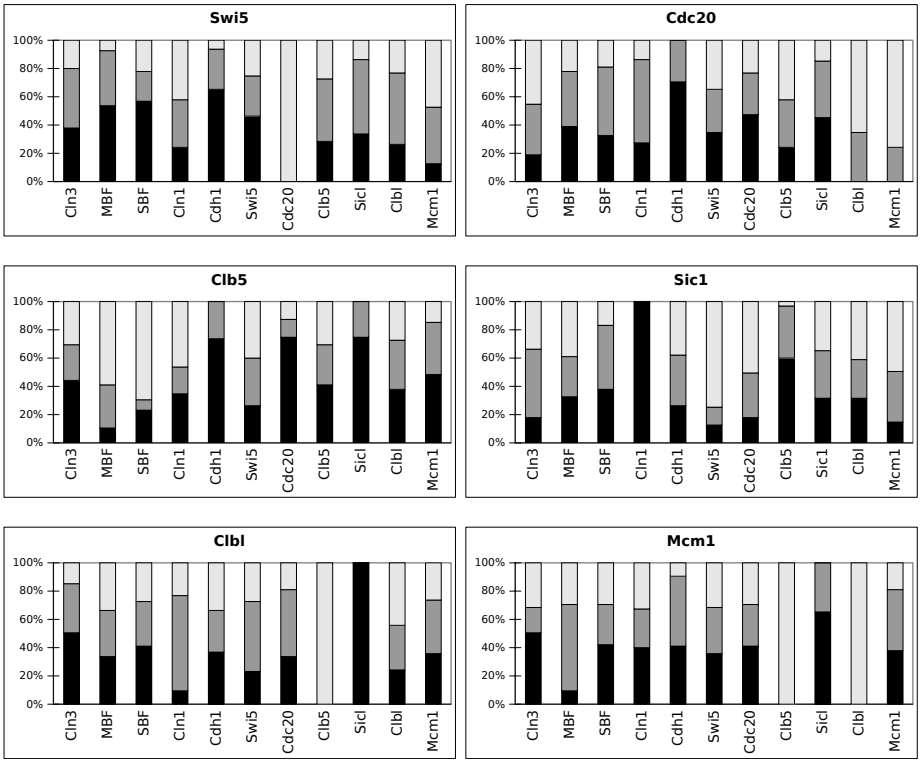

Fig. 6: Results for the genes Swi5, Cdc20, Clb5, Sic1, Clb1 and Mcm1.

6 Discussion

By looking at Fig. 5 and 6, it is interesting to note that, in some cases, the statistics of the networks were capable of almost excluding one possibility of relationship - as shown in Swi5 → Cln3, SBF → Clb5, MBF → Mcm1, and others - transforming some connections from undetermined into partially determined connections. These results show that the cell cycle pathway constrains some connections, therefore restricting the whole network [12].

We can attribute this phenomenon to the high dependency that the determination of a network connection has on other connections. The three steps of the presented algorithm perform a search over the space of possibilities of the influence of a set of genes over a single gene. If one of these influences is *a priori* determined (or known), this result can bias other connections. For example, let us suppose that genes A and B have to produce a positive output over a gene C, according to some restriction imposed by the time-series data. If we already know that gene A has no relationship to gene C, gene B must have a positive relationship to gene C.

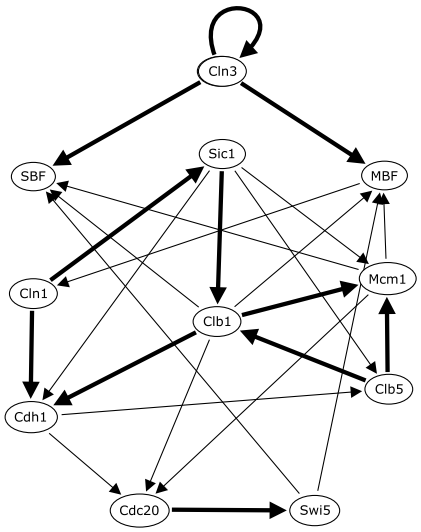

Fig. 7: The determined (bold arrows) and partially determined connections (light solid arrows) inferred by the consecutive application of the three steps of the algorithm.

Therefore, this high dependency on the determination of a connection over the network makes the use of Fig. 5 and 6 very restricted. If we simply use a relationship with a high weight to be our "best guess" on the connection between two genes, this choice can constrain other relationships, leading the system to a more or less determined state, or even creating a network that is not consistent with the data.

We can say that Fig. 5 and 6 represent a good approximation of a "greedy" heuristic for finding one network. It can be done in the following way. Firstly, calculate the frequency of the connections of a set of consistent networks. Secondly, choose the most determined connection to be fixed with the relationship that has the greater weight. Thirdly, recalculate the set of networks and return to step one.

Another fact to be pointed out is the importance of the inferred partially determined connections. Although these connections can not be directly used to construct a network like the determined connections, it can guide some biological experiments, since a partially deterministic connection states that at least one type of relationship between two genes is not possible. We could use the frequencies generated in Fig. 5 and 6 to attribute a *strength of connection* to the relationship of a partially determined connection, e.g., the interference of Clb1 on SBF can be stated as 80% (or a probability of 0.8) of being an inhibition.

A closer look into the statistics raises also an interesting question: the network chosen by the nature would not be easily detectable? Or even better: would not the collected data be enough to constrain Fig. 5 and 6 into nature's choice? We could answer this question by pointing out a piece of information that makes a

huge difference between our model and nature's choice: the chemical interactions between proteins. Evidently, some of the connections considered on many steps of the algorithm here presented can not exist due to chemical incompatibilities. In some sense, nature has more information to constrain its network than we do.

7 Conclusion and Future Research

This paper proposes an algorithm to perform analyses for discovering gene regulatory interactions from time-series data under the Boolean network model. In fact, the inference of gene regulatory networks is a one-to-many inverse problem in the sense that there may exist several networks consistent with the dataset. In order to analyze the gene interactions, we have generated several networks and considered only the consistent ones. We have applied our methodology to an artificial dataset that had been generated by a Boolean network that models the budding yeast cell cycle [14]. By this application, we have shown that this analysis of gene interactions could be a first step for gene relationships detection with a high flexibility to include biological knowledge.

A challenge always presented in any gene regulatory model is its usefulness. It would be very interesting if a model could help biological experiments in understanding gene interactions. The model here presented is capable of inferring some of these connections from time-series data of gene expressions, and this inference process is helped by all *a priori* knowledge available.

Hence, an interesting feature to be added to our method would be the ability to indicate which connection should be verified in the wet lab to help determine others. As stated in the last section, the network connections are very dependent of each other, and the determination of one connection could constrain the whole network. What we envisage with our method is a model that points out which connections should be determined in the wet lab that in turn would constrain as many other connections as possible and consequently could facilitate some biological experiments. We are investigating the possibility to put our algorithm in the context of a *constraint solving problem* (CSP) [15]. There are CSP solver techniques that may help us to analyze the gene interactions as we did in this paper.

However, there are other characteristics to be sought that could constrain the network towards nature's choice. One feature not explored in this paper is the dynamical aspects of the network. There are indications, as stated by Kauffman [10], that nature would prefer networks with a small quantity of attractors - the gene pattern expression that leads the system to itself- and large basins of attraction - the set of gene pattern expressions that leads the system to one attractor. The network constructed by Li et al. [14] has these characteristics. Therefore, a connection statistics calculated only from networks with a few number of attractors - or other dynamical characteristic - could create a well established result.

Concluding, we think that the model here presented is a remarkable first step of the construction of a system to infer gene interactions. Our intention now is

to test this procedure with another artificial data and, perhaps, biological data also; and to implement some topics presented in this section. We understand that any inference procedure can not have success if it does not contain biological and computational expertise, therefore the future steps of this research have to be centered on the difficulties of a wet lab, or its limitations.

References

1. Kauffman, S. A.: Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology. 22, 437-467 (1969)
2. Shmulevich, I., Dougherty, E. R., Kim, S., Zhang, W.: Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics. 18(2), 261-274 (2002)
3. Zhang, S. Q., Ching, W. K., Ng, M. K., Akutsu, T.: Simulation study in Probabilistic Boolean Network models for genetic regulatory networks. 1(3):217-40 (2007)
4. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian networks to analyze expression data. Journal of Computational Biology. 7(3-4), 601-620 (2000)
5. Goodwin, B. C.: Temporal Organization in Cells; A Dynamic Theory of Cellular Control Process. Academic Press (1963)
6. Karlebach, G., Shamir, R.: Modelling and analysis of gene regulatory networks. Nature. 9, 770-780 (2008)
7. Hartemink, A. J.: Reverse engineering gene regulatory networks. Nature. 23(5), 554-555, (2005)
8. Hecker, M., Lambeck, S., Toepfer, S., Someren, E. van, Guthke, R.: Gene regulatory network inference: Data integration in dynamic models - A review. BioSystems. 96, 86-103 (2009)
9. Shmulevich, I., Dougherty, E. R., Zhang, W.: From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks. Proceedings of the IEEE. 90, 1778-1792 (2002)
10. Kauffman, S. A.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford Univ. Press. (1993)
11. Huang, S.: Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery. J. Mol. Med.. 77, 469-480 (1999)
12. Lau, K. Y., Ganguli, S., Tang, C.: Function Constrains Network Architecture and Dynamics: A Case Study on the Yeast Cell Cycle Boolean Network. Physics Review E. 75(5), 1-9 (2007)
13. Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., Futcher, B.: Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomices cerevisiae* by Microarray Hybridization. Molecular Biology of the Cell. 9, 3273-3297 (1998)
14. Li, F., Long, T., Lu, T., Ouyang, Q., Tang, C.: The Yeast Cell-Cycle Network is Robustly Designed. PNAS of the USA. 101(14), 4781-4786 (2004)
15. Tsang, E.: Foundations of Constraint Satisfaction. Academic Press (1993)