BioSystems 115 (2014) 23–32

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Dynamical and topological robustness of the mammalian cell cycle network: A reverse engineering approach

Gonzalo A. Ruz[a,*], Eric Goles[a], Marco Montalva[a], Gary B. Fogel[b]

[a] *Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Santiago, Chile*
[b] *Natural Selection, Inc., 5910 Pacific Center Boulevard, Suite 315, San Diego, CA 92121, USA*

ARTICLE INFO

Article history:
Received 23 September 2013
Received in revised form 28 October 2013
Accepted 30 October 2013

Keywords:
Gene regulatory networks
Boolean networks
Threshold networks
Update robustness
Topology robustness
Bees algorithm

ABSTRACT

A common gene regulatory network model is the threshold Boolean network, used for example to model the *Arabidopsis thaliana* floral morphogenesis network or the fission yeast cell cycle network. In this paper, we analyze a logical model of the mammalian cell cycle network and its threshold Boolean network equivalent. Firstly, the robustness of the network was explored with respect to update perturbations, in particular, what happened to the attractors for all the deterministic updating schemes. Results on the number of different limit cycles, limit cycle lengths, basin of attraction size, for all the deterministic updating schemes were obtained through mathematical and computational tools. Secondly, we analyzed the topology robustness of the network, by reconstructing synthetic networks that contained exactly the same attractors as the original model by means of a swarm intelligence approach. Our results indicate that networks may not be very robust given the great variety of limit cycles that a network can obtain depending on the updating scheme. In addition, we identified an omnipresent network with interactions that match with the original model as well as the discovery of new interactions. The techniques presented in this paper are general, and can be used to analyze other logical or threshold Boolean network models of gene regulatory networks.

1. Introduction

Over forty years ago, Stuart Kauffman introduced Boolean networks (BNs) as a mathematical model of gene regulatory networks (GRNs) (Kauffman, 1969). GRNs represent the process of gene regulation, which determines when and where genes will be active/inactive through the interactions of DNA, RNA, proteins, and other substances within the cell. BNs are very simple and can be described as follows. Nodes represent genes and edges represent the interaction between the genes (i.e., a regulation process). Each gene is considered to act as an on-off device, the two states (on/off) represent respectively, the status of a gene being active (gene value = 1) or inactive (gene value = 0). Given that each node can only have two values, for a network with n nodes, this implies that the network has 2^n different states. The dynamics of the network (how the values of the nodes change through time) are governed by a set of Boolean rules and an updating scheme. In the original model, the updating scheme was considered to be synchronous or parallel, such that at each time step, node values for all nodes were updated at the same time. An important characteristic of BNs are steady state attractors for network convergence. There are two types of attractors: (1) fixed point, where once a network reaches that state it can never escape, and (2) a limit cycle, where the network returns to a previous state with a certain periodicity. The attractors are of interest in the context of GRNs since they represent different cell types.

BNs are very popular within GRN modelers, in part due to their simplicity. However, this same characteristic is a focus for criticism as not being very realistic. For example, parallel updating schemes have been used to model the *Arabidopsis thaliana* floral morphogenesis network (Mendoza and Alvarez-Buylla, 1998), the fission yeast cell cycle network (Davidich and Bornholdt, 2008), and the budding yeast cell cycle network (Li et al., 2004). These clearly include a large assumption about the extreme regularity and tight control of global gene expression. At first sight, this could lead to think that the parallel updating scheme is unrealistic and wrong. Nevertheless, it is capable of exhibiting a dynamic behavior similar to that of biological cells (Kauffman et al., 2003; Wuensche, 2004), specially when cell differentiation is associated to fixed points, which are invariant to update perturbation, thus, making the parallel updating mode a computationally convenient selection, which is quite exhaustive in the presence of strong stability, where the updating scheme does not influence significantly the dynamic of the model. An example where this phenomena occurs, and that one could consider that the parallel is exhaustive enough, given that the behavior exhibited with the parallel is similar for any other deterministic updating

* Corresponding author. Tel.: +56 223311200.
E-mail addresses: gonzalo.ruz@uai.cl, gonzalo.ruz@gmail.com (G.A. Ruz).

0303-2647/$ – see front matter © 2013 Elsevier Ireland Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.biosystems.2013.10.007

scheme is the budding yeast cell cycle network (Li et al., 2004) which contains seven fixed points, and the basin of attraction for each fixed point, in particular the one that represents the G1 phase, does not change significantly when changing the updating scheme (Goles et al., 2013). For the general case where one can not assure strong stability beforehand, an interesting question arises: what happens to the models that assume a parallel updating scheme if a change in the updating scheme (an update perturbation) occurs? Do the attractors remain the same, or are new attractors derived?

The construction of GRN models from data is typically referred to as a reverse engineering problem (Liang et al., 1998; Akutsu et al., 1999). Building GRNs is a difficult task given the large space of possible GRN models that might fit the data and the need to search that space in reasonable time to derive useful solutions. Several approaches using evolutionary computation (EC) have been proposed to aid in this search. For example in Mendoza et al. (2012), Boolean network models of GRN were inferred using genetic algorithms (GAs) to optimize a Tsallis entropy function. GAs were also used in Repsilber et al. (2002) for the reconstruction of multistate discrete network models for GRN, allowing each node (gene) to have more than two states, and also in Kikuchi et al. (2003) for modeling GRN as an S-system. Differential evolution (DE) has been used for GRN reconstruction using S-systems in Chowdhury et al. (2012), and in Noman and Iba (2007) with an information criteria-based fitness evaluation instead of the conventional mean squared error-based fitness evaluation. Other optimization methods such as simulated annealing (SA) have been used. In Liu et al. (2009), SA was used to model GRNs as Bayesian networks, and Gonzalez et al. (2007) used SA to derive S-system models of biochemical networks, whereas Ruz and Goles (2010) used threshold Boolean networks. Swarm intelligence has also been used for the inference of GRNs. For instance, in Kentzoglanakis and Poole (2012) a combination of particle swarm optimization (PSO) and ant colony optimization (ACO) was used to reverse engineer GRNs, under the recurrent neural network (RNN) model, from temporal gene expression data. The ACO has also been used to search for network structures with PSO used to finding the RNN model parameters. Similarly, in Xu et al. (2007), PSO was used to find the network structure and parameters of GRN modeled by RNN, using time-series gene expression data. A comparison between EAs, PSO, and an artificial bee colony (ABC) approach, with GRNs modeled as S-systems, was conducted in Forghany et al. (2012). The results on two small-size and a medium-sized hypothetical gene regulatory networks showed that a modified version of ABC outperformed the other techniques. Recently, the bees algorithm (BA) (Pham et al., 2006) for reverse engineering of GRN was introduced in Ruz and Goles (2013). Comparisons with SA for learning threshold Boolean networks showed that the bees algorithm outperformed SA, obtaining a larger number of solutions using fewer edges in the network. The bees algorithm has been used to build synthetic networks of the budding yeast cell-cycle in Ruz et al. (2012), and for promoting cell proliferation for biotechnological applications. In Ruz and Goles (2012), a reverse engineering technique was applied to the reconstruction of the mammalian cell cycle network using the binary gene expression data generated by the logical model in Fauré et al. (2006). This reverse engineering method used an information theoretic approach combined with a modified version of the original bees algorithm.

Here we present extensions to Ruz and Goles (2012). First, we analyze the dynamics of the mammalian cell cycle network under different updating schemes. It is important to note that, while in Ruz and Goles (2012) we provided results for only 5000 sequential updating schemes, in this paper, we analyzed all possible deterministic updating schemes. This is a difficult problem, given that there are an exponential number of updates. If the network has n nodes, the number of updates is given by Demongeot et al. (2008):

$$T_n = \sum_{k=0}^{n-1} \binom{n}{k} T_k, \qquad T_0 = 1.$$

For a mammalian cell cycle network with $n = 10$, we have that $T(10) = 102{,}247{,}563$. To analyze this vast amount of dynamics, we combined mathematical results with recent computational techniques developed in Goles et al. (2013) and in Aracena et al. (2013). We then analyzed the topology of the network, and for this portion, we used the bees algorithm to reconstruct synthetic networks that contained exactly the same attractors of the original model. Using this reverse engineering approach, we are able to identify interactions which are always present and that match with the original model as well as new interactions in these GRNs.

The rest of the paper is organized as follows. Section 2 gives a brief description of Boolean networks, the mammalian cell cycle network of Fauré et al. (2006), and the bees algorithm. The robustness of the network under all the deterministic updating schemes is carried out in Section 3. Section 4 approaches the study of the topology robustness using the bees algorithm. General conclusions are offered in Section 5.

2. Background

2.1. Boolean networks

Let $\mathbf{x}$ be a finite set of n variables, $\mathbf{x} = \{x_1, \ldots, x_n\}$, with $x_i \in \{0, 1\}$ for $i = 1, \ldots, n$. A Boolean network is a pair (G, F), where $G = (\mathbf{V}, \mathbf{E})$ is a finite directed graph; $\mathbf{V}$ being the set of n nodes and $\mathbf{E}$ the set of edges. F is a Boolean function, $F: \{0, 1\}^n \to \{0, 1\}^n$ composed of n local functions $f_i: \{0, 1\}^n \to \{0, 1\}$. Furthermore, each local function f_i depends only on variables belonging to the neighborhood $V_i = \{j \in \mathbf{V} | (j, i) \in \mathbf{E}\}$. The indegree, K, of vertex i is $|V_i|$. The updating schemes are repeated periodically, and since the hypercube is a finite set, the dynamics of the network converges to attractors which are fixed points or limit cycles, defined by

- Fixed point: $x_i^{t+1} = x_i^t$ for $i = \{1, \ldots, n\}$.
- Limit cycle: $x_i^{t+p} = x_i^t$ for $i = \{1, \ldots, n\}$.

where $p > 1$ is a positive integer called the limit cycle length. The set of states that can lead the network to a specific attractor is termed the basin of attraction. There are many ways of updating the values of a Boolean network, some examples are (Aracena et al., 2009):

- Parallel or synchronous mode: where every node is updated at the same time.
- Sequential updating mode: where in every time step, every node is updated in a defined sequence.
- Block-sequential: the set of nodes, for a given sequence, is partitioned into blocks. The nodes in a same block are updated in parallel, but blocks follow each other sequentially.
- Asynchronous deterministic: where in every time step, only one node is updated following a defined sequence.

2.2. The mammalian cell cycle model

Initially, a differential model for the control of the mammalian cell cycle network was introduced by Novak and Tayson (2004). Then, a logical version for this model was presented by Fauré et al. (2006). In this logical model, the mammalian cell cycle network consisted of 10 genes (that characterize enzymatic complexes and cofactors), therefore, there are $2^{10} = 1024$ possible states. When

$$W = \left(\begin{array}{c|cccccccccc} & CycD & Rb & E2F & CycE & CycA & p27 & Cdc20 & Cdh1 & UbcH10 & CycB \\ CycD & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Rb & -4 & 0 & 0 & -1 & -1 & 3 & 0 & 0 & 0 & -4 \\ E2F & 0 & -3 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -3 \\ CycE & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ CycA & 0 & * & * & 0 & * & 0 & * & * & * & 0 \\ p27 & -2 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & -2 \\ Cdc20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ Cdh1 & 0 & 0 & 0 & 0 & -1 & 1 & 5 & 0 & 0 & -3 \\ UbcH10 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 3 & 1 \\ CycB & 0 & 0 & 0 & 0 & 0 & 0 & -3 & -1 & 0 & 0 \end{array}\right) \qquad \Theta = \begin{pmatrix} 0 \\ -1 \\ -1 \\ 0 \\ * \\ -1 \\ 0 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

Fig. 1. Weight matrix and the threshold vector. * represents no parameters found.

updated in parallel, the network has two attractors. One attractor is a limit cycle of length seven that is used to model the cell cycle, and the other is a fixed point attractor. The updating rules for each node are as follows:

$$CycD = CycD \tag{1}$$

$$Rb = (\overline{CycD} \wedge \overline{CycB}) \wedge ([\overline{CycE} \wedge \overline{CycA}] \vee p27) \tag{2}$$

$$E2F = (\overline{Rb} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{Rb} \wedge \overline{CycB}) \tag{3}$$

$$CycE = (E2F \wedge \overline{Rb}) \tag{4}$$

$$CycA = (\overline{Rb} \wedge \overline{Cdc20} \wedge \overline{Cdh1 \wedge UbcH10}) \wedge (E2F \vee CycA) \tag{5}$$

$$p27 = (\overline{CycD} \wedge \overline{CycB}) \wedge ([\overline{CycE} \wedge \overline{CycA}] \vee [p27 \wedge \overline{CycE \wedge CycA}]) \tag{6}$$

$$Cdc20 = CycB \tag{7}$$

$$Cdh1 = (\overline{CycA} \wedge \overline{CycB}) \vee Cdc20 \vee (p27 \wedge \overline{CycB}) \tag{8}$$

$$UbcH10 = \overline{Cdh1} \vee (Cdh1 \wedge UbcH10 \wedge [Cdc20 \vee CycA \vee CycB]) \tag{9}$$

$$CycB = \overline{Cdc20} \wedge \overline{Cdh1} \tag{10}$$

Note that some of the previous equations have been modified with respect to those of Fauré et al. (2006) but remain equivalent. More recently, an attempt to reconstruct a threshold Boolean network model exhibiting the same dynamics as Fauré et al. (2006) was conducted in Ruz and Goles (2012), showing that *CycA* cannot be modeled by a single threshold function given its non-linear nature. The resulting parameters (weights and thresholds) and structure for the mammalian cell cycle threshold Boolean network appears in Figs. 1 and 2 respectively. The updates of each node are computed by

$$x_i(t+1) = f_i(\mathbf{x}) = u\Big(\sum_{j=1}^{n} \omega_{ji} x_j(t) - \theta_i\Big) \tag{11}$$

$$u(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \tag{12}$$

with ω_{ji} the weight of the edge coming from node j into the node i, and θ_i the activation threshold of node i.

One can see in Fig. 1 that no parameters were found for *CycA* that would replicate the dynamics given by the logical rules. It was found that this could only be achieved by adding an additional layer with two nodes connecting to *CycA*. By adding two extra nodes, the attractors of the original model were changed, therefore, for *CycA*, the logical rules were used to compute the dynamics. With the resulting (almost) threshold Boolean network, the dynamics using a parallel updating scheme was the same as the one obtained in the original model, with two attractors, one fixed point and one limit cycle of length seven. This can be visualized in the state transition graph shown in Fig. 3.

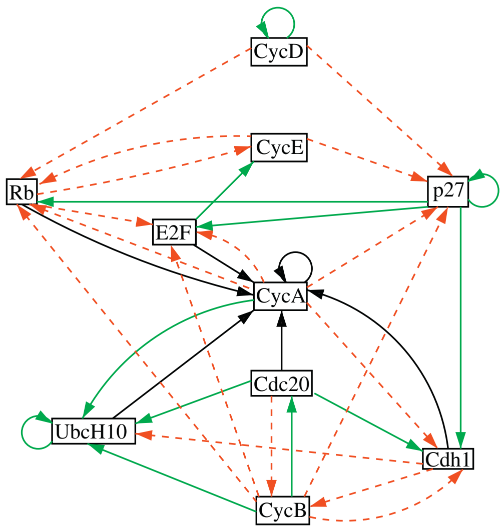

Fig. 2. The structure of the mammalian cell cycle threshold Boolean network. The green/solid edges represent positive weights (activations), the red/dashed edges represent negative weights (inhibitory), and no weight values were found for the black/solid edges. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.3. The bees algorithm

The BA is a population-based search algorithm for function optimization and combinatorial optimization problems. It was first introduced in Pham et al. (2006), and studied and compared to other meta-heuristic algorithms in Pham and Castellani (2009). The algorithm is based on honeybee foraging. In the BA, each bee represents a candidate solution, a flower patch represents a local search area, and the amount of food the bee collects from the flower patch is the fitness value. The parameters of the BA that must be specified by the user are shown in Table 1 and the flowchart of the algorithm appears in Fig. 4.

The BA initiates with a random population of scout bees, as candidate solutions for the optimization problem. The fitness of each scout bee is measured. Scout bees are then ranked in descending order according to their fitness value. The first *ne* scout bees are

Table 1
The bees algorithm parameters.

ns	Number of scout bees
ne	Number of elite sites
nb	Number of best sites
nre	Recruited bees for elite sites
nrb	Recruited bees for best sites
maxi	Maximum number of iterations

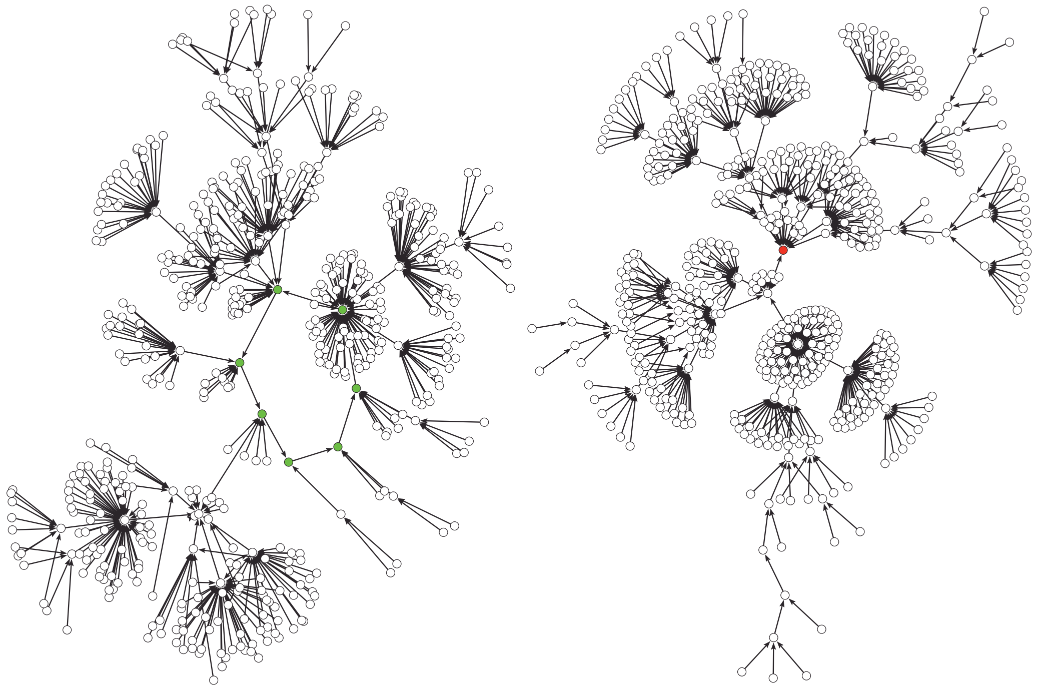

Fig. 3. State transition graph of the mammalian cell cycle network using the parallel updating scheme. The red circle represents the fixed point state, the seven green circles represent the states that belong to the limit cycle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

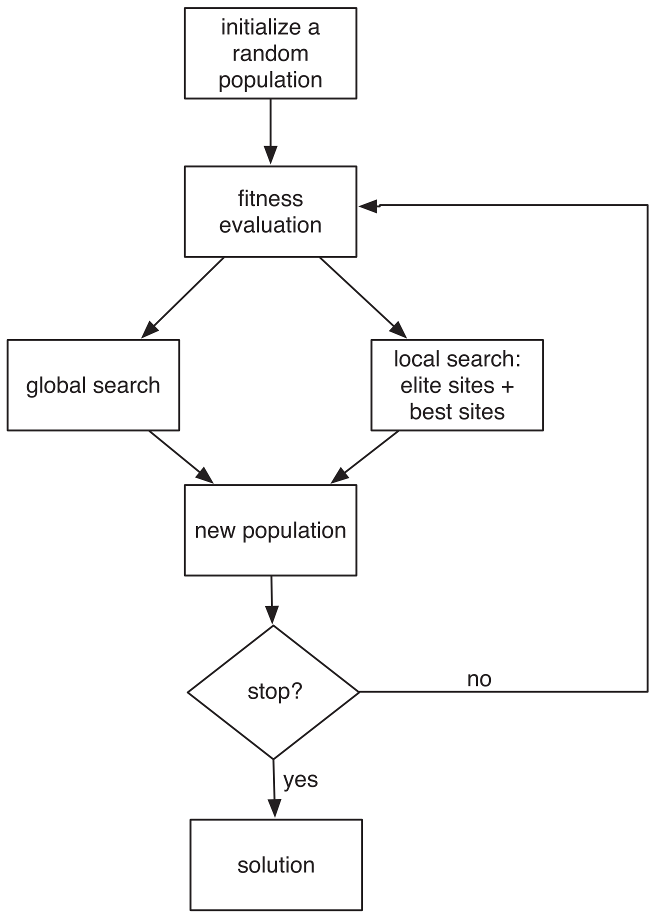
initialize a random population
fitness evaluation
global search
local search: elite sites + best sites
new population
stop?
no
yes
solution

Fig. 4. The bees algorithm flowchart.

considered as elite sites and *nre* bees are recruited to perform local search at these elite sites. Subsequent *nb* scout bees by decreasing fitness are considered as best sites and *nrb* bees are recruited to perform local search at these best sites. Then $ns - (ne + nb)$ bees are left for random search. The bees with highest fitness from each elite, best, and random sites are introduced into the scout bees population. The scout bees are re-ranked and only the top *ns* bees are considered, the rest are discarded, this generates a new population. The scout bee with the highest fitness from the newly formed population is checked to see if it satisfies any process termination criteria. If a termination condition is not met, then the new population is used to repeat the process of local and global search until the termination condition is met or *maxi* has been reached.

3. Dynamical robustness of the mammalian cell cycle network

Changes in the updating scheme can affect the limit cycles. It is straightforward to show that fixed points are not affected by changes in the updating scheme, therefore, in this section we will analyze what happens to the mammalian cell cycle network under update perturbations, in particular, what happens to the limit cycle attractor found in the parallel updating scheme when changed to all the possible deterministic updating schemes. To carry out the study of the complete dynamics of the network, we separated the analysis in two cases; $CycD = 1$ and $CycD = 0$. Since node *CycD* remains constant through all the mammalian dynamics, what happens when $CycD = 1$ will be completely independent of the other case (i.e., the whole dynamic can be separated in two sets of state configurations; $A = \{x \in \{0, 1\}^{10} : CycD = 1\}$ and $B = \{x \in \{0, 1\}^{10} : CycD = 0\}$ (note that $|A| = |B| = 512$) such that transitions between elements of *A* to *B* (or vice-versa) do not exist). Thus, in the first part ($CycD = 1$), we simplified the network in order to prove a general dynamical property of the mammalian network which is valid for any deterministic updating mode (Proposition 3.1) that says that, at least half of the mammalian dynamic (those elements of *A*), updated in every

Table 2
Attractor analysis for all the dynamics of the subnetwork in Fig. 5. In this case, there is only one limit cycle per dynamic, with attraction basin of size 512 (including the limit cycle configurations). $|D_i|$ is the number of dynamics whose only limit cycle has length *i*.

Different dynamics	$\|D_2\|$	$\|D_3\|$	$\|D_4\|$	$\|D_5\|$	$\|D_6\|$	$\|D_7\|$
1696	16 (0.9%)	298 (17.6%)	812 (47.9%)	432 (25.5%)	132 (7.8%)	6 (0.4%)

deterministic way, does not converge to fixed point attractors (i.e. they converge only to limit cycles).

3.1. Case 1: *CycD* = 1

We start by making some simplifications to the mammalian network of Fig. 2. For this, note the following implications:

- If $CycD=1$, then $Rb=0$ in (2) and $p27=0$ in (6).
- If ($Rb=0 \wedge p27=0$), then $E2F=\overline{CycA}\wedge\overline{CycB}$ in (3).
- If $Rb=0$, then $CycE=E2F$ in (4).
- If $Rb=0$, then $CycA=(\overline{Cdc20}\wedge\overline{Cdh1\wedge UbcH10})\wedge(E2F\vee CycA)$ in (5).
- If $p27=0$, then $Cdh1=(\overline{CycA}\wedge\overline{CycB})\vee Cdc20$ in (8).

Thus, in at least one iteration step, the dynamical behavior of the mammalian network can be described by the dynamic of the subnetwork showed in Fig. 5 and with the following local functions on its nodes:

$$E2F=\overline{CycA}\wedge\overline{CycB} \quad (13)$$

$$CycE=E2F \quad (14)$$

$$CycA=(\overline{Cdc20}\wedge\overline{Cdh1\wedge UbcH10})\wedge(E2F\vee CycA) \quad (15)$$

$$Cdc20=CycB \quad (16)$$

$$Cdh1=(\overline{CycA}\wedge\overline{CycB})\vee Cdc20 \quad (17)$$

$$UbcH10=\overline{Cdh1}\vee(Cdh1\wedge UbcH10\wedge[Cdc20\vee CycA\vee CycB]) \quad (18)$$

$$CycB=\overline{Cdc20}\wedge\overline{Cdh1} \quad (19)$$

Using this simplification, we have the following proposition:

Proposition 3.1. *Let S be the set of all the deterministic updating schemes of the mammalian network of Fig.2. Then, CycD = 1 ⇔ ∀ s ∈ S, the mammalian dynamics, generated by the nine remaining nodes, has no fixed point.*

Proof. ⇐) if $CycD=0$, then we know from Fauré et al. (2006), that the mammalian dynamic, generated by the nine remaining nodes, has one fixed point for the parallel updating scheme.

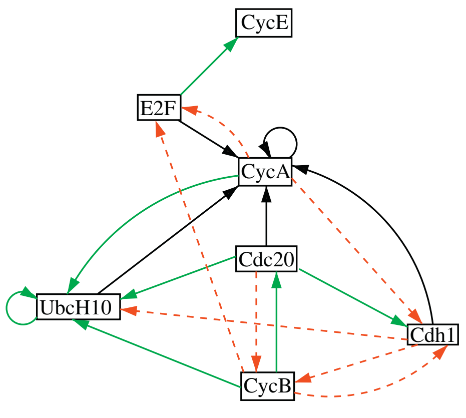

Fig. 5. A subnetwork of Fig. 2 obtained when *CycD* = 1.

Table 3
General attractor analysis for all the dynamics of the subnetwork in Fig. 2 when *CycD* = 0.

	Number of dynamics	Range of limit cycles length
Different	13,0297	2–8
With 1 limit cycle	19,719 (15.1%)	2–8
With 2 limit cycles	163 (0.1%)	2–6
Without limit cycles	110,415 (84.7%)	–

⇒) Let $CycD=1$ and $s\in S$. We know by the above implications, that in at most one iteration step, $Rb=0$ and $p27=0$. Therefore, the mammalian dynamics generated by the seven remaining nodes can be described by the dynamics of the subnetwork shown in Fig. 5. Let N be such a subnetwork, we will prove that N has no fixed point by contradiction. Let us consider that $y'=(E2F, CycE, CycA, Cdc20, Cdh1, UbcH10, CycB)\in\{0,1\}^7$ is a fixed point of N. We can consider two possibilities; $Cdc20=1$ or $Cdc20=0$ and we will have the following implications for each:

- If $Cdc20=1$, then $CycB=1$ in (16) but, also, if $Cdc20=1$, then $CycB=0$ in (19), which is a contradiction.
- If $Cdc20=0$, then $CycB=0$ in (16). If ($Cdc20=0\wedge CycB=0$), then $Cdh1=1$ in (19). If ($Cdc20=0\wedge CycB=0\wedge Cdh1=1$), then $CycA=0$ in (17). If ($Cdc20=0\wedge Cdh1=1\wedge CycA=0$), then ($E2F=0\vee UbcH10=1$) in (15) but, on the other hand, if we consider the above values of; $CycA=0$, $CycB=0$, $Cdc20=0$ and $Cdh1=1$, we will have that $E2F=1$ and $UbcH10=0$ in (13) and (18) respectively, which is a contradiction.

Therefore, N cannot have fixed points. □

Next we performed a more detailed study of all the attractors for this case $CycD=1$ (which can only be limit cycles, by the above proposition) with the algorithms developed in Aracena et al. (2013) for the subnetwork of Fig. 5. These results are summarized in Table 2.

Table 2 shows, amongst other things, that in addition to the limit cycle obtained for the synchronous update in Fauré et al. (2006), there could also appear other 5 limit cycles with length 7. However, our simulations show that:

- Only 3 limit cycles of length 7 are really different. In other words, under the presence of *CycD* and for any (deterministic) updating mode considered, the mammalian dynamic has only 1 limit cycle of length 7; that of Fauré et al. (2006) for the synchronous update or some of those showed in Fig. 6.
- From the 3 limit cycles above, we observe that other simple transitions appears; the inhibitions of nodes *UbcH*10, *Cdh*1 and *Cdc*20.
- Configurations 1011000100, 1000100011 and 1000101011 are common in all the previous limit cycles.
- The other configurations differ by a Hamming distance of at most 2.

3.2. Case 2: *CycD* = 0

When $CycD=0$ the resulting network has nine nodes and the results are summarized in Table 3. Note that most of the dynamics (≈85%) have a single attractor; a fixed point. In the rest of the dynamics (≈15%), only one limit cycle appears (with a maximum

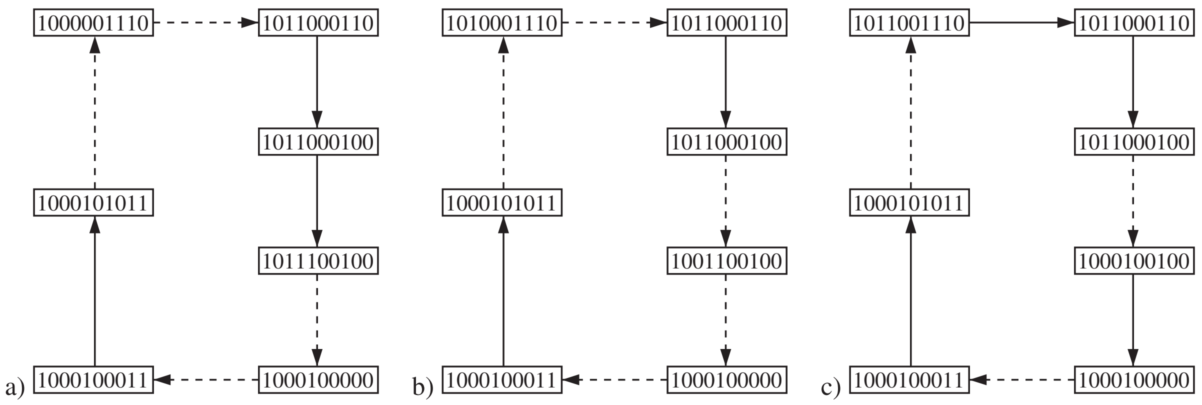

Fig. 6. a) Limit cycle for the update schedule (*CycD*, *Rb*, *E2F*, *CycA*, *p*27, *Cdc*20, *Cdh*1, *UbcH*10, *CycB*)(*CycE*). b) Limit cycle for the update schedule (*CycD*, *Rb*, *CycE*, *CycA*, *p*27, *Cdc*20, *Cdh*1, *UbcH*10, *CycB*)(*E2F*). c) Limit cycle for the update schedule (*CycA*, *p*27, *Cdc*20, *Cdh*1, *UbcH*10, *CycB*)(*Rb*, *E2F*)(*CycE*)(*CycD*).

length of 8), except for a few ones (≈0.1%) where two additional limits cycles appears (but the maximum length decreases to 6). Details of the dynamics that have limit cycles are shown in Table 4. The most frequent limit cycle lengths are 2 and 4 whereas the least frequent are 7 and 8, however, the latter have the maximum average basin sizes whereas the former have the minimum average basin sizes.

In summary, the previous cases of analysis allow us to say that every deterministic dynamic of the mammalian network can be divided into two components; one containing half of the configurations converging to a single limit cycle (case $CycD = 1$), the other one, with a single fixed point and, possibly, with at most two limit cycles.

Two different dynamics, using block sequential updates, of the mammalian network are shown in Figs. 7 and 8. In Fig. 7, the state transition graph shows that the network converges to three attractors: the unique fixed point (with $CycD = 0$), a limit cycle of length 8 (with $CycD = 0$), and a limit cycle of length 4 (with $CycD = 1$). In Fig. 8, the state transition graph shows that the network converges to four attractors: the unique fixed point (with $CycD = 0$), a limit cycle of length 2 (with $CycD = 0$), a limit cycle of length 6 (with $CycD = 0$), and another limit cycle of length 6 (with $CycD = 1$).

Although we have analyzed all the deterministic dynamics, it should be noted that not all of these are biologically meaningful nor even biologically feasible. As pointed out in Qu et al. (2003), there have been many models that nominally replicate the dynamics of the cell cycle under specific conditions, but no unified theory of cell cycle dynamics has emerged. One approach (Qu et al., 2003) is to analyze the key regulators in the control of the G1-to-S transition and then the G2-to-M transition, before linking them together and analyzing the whole process. Cyclin-dependent kinase (Cdk) network models that control successive phases of the mammalian cell cycle have been proposed in Gérard and Goldbeter (2009) and Gérard and Goldbeter (2011). In both cases, it is shown that the ordered sequential, repetitive activation of the various cyclin/Cdk complexes ensures proper progression along the successive phases of the cell cycle. In particular, the oscillations are based on the sequential activation of cyclin D/Cdk4-6, cyclin E/Cdk2, cyclin A/Cdk2 and cyclin B/Cdk1 and with the property that, once activated, each of these complexes inactivates the preceding cyclin/Cdk complex in the network.

Table 4
A more detailed analysis of the values showed in Table 3 for the dynamics having a limit cycle.

	Dynamics with at least 1	Average basin size
Limit cycle of length 2	6587	140.5
Limit cycle of length 3	3383	321.7
Limit cycle of length 4	6148	273.4
Limit cycle of length 5	2892	298.1
Limit cycle of length 6	919	325
Limit cycle of length 7	21	363.3
Limit cycle of length 8	63	372.9

4. Topology robustness of the mammalian cell cycle network

In Ruz and Goles (2012), we used a combination of mutual information and the BA to reconstruct a threshold Boolean network which exhibited the same dynamics as the logical model in Fauré et al. (2006). However, here we relaxed the problem, and considered the search for threshold Boolean networks that shared exactly the same attractors, and no others, of Fauré et al. (2006), but not necessarily the same complete dynamics. The resulting synthetic networks provided an indication of the robustness of the network topology, in the sense of which edges are always present and which can vary, without losing the attractors or generating new ones. Also, as noted in Ruz and Goles (2012), *CycA* cannot be modeled by a single threshold function, therefore, it is natural to ask if this phenomena is due to the specific attractors or the complete dynamics of the logical model. To carry out this analysis we used BA to search for networks that contain exactly the same attractors as the original model and then analyzed the resulting topologies.

There are three parts that need to be defined when implementing the BA.

4.1. Coding of solutions

The solutions, network weights and thresholds, are coded using an adjacency matrix Ω and a vector Θ respectively, as in Fig. 1. The initial Ω_g, $g = 1, \ldots, ns$ in the BA were generated randomly, with an indegree $K \leq 5$ and integer weight values in the range $[-5, 5]$. The threshold vectors Θ_g where all set to the original values that appear in Fig. 1. The * was replaced by a 0.

4.2. Definition of the fitness function

The fitness function for the Boolean network B, was given by the deviation of the network output, defined by o_i for each node i, and the target value a_i (attractor) for each node i, which was computed by

$$fitness(B) = \frac{1}{8n}\sum_{t=1}^{8}\sum_{i=1}^{n}(o_i(t) - a_i(t))^2 \tag{20}$$

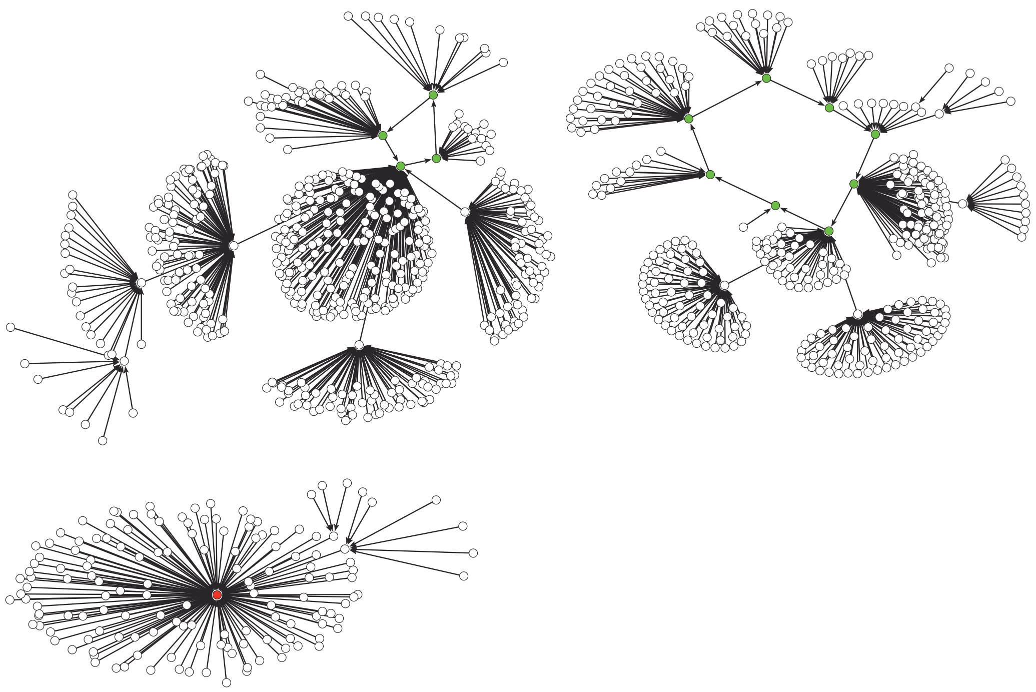

Fig. 7. State transition graph of the mammalian cell cycle network with the following updating scheme: (*CycD*, *Rb*, *Cdc*20, *Cdh*1, *CycA*)(*p*27, *UbcH*10, *CycB*)(*E2F*)(*CycE*). The red circle represents the fixed point state, the 12 green circles represent the states that belong to two limit cycles, one of length 8 and another of length 4. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where *n* is the number of nodes in the network, and 8 is the number of attractor vectors that the network must contain. For this particular application, there is one limit cycle of length seven and one fixed point. Also, the solutions were penalized by adding half of the maximum value of (20), (i.e. 0.5, when additional attractors appeared).

4.3. Neighborhood search strategy

During the local search, each recruited bee from the elite or best sites generates a candidate solution, Ω_{new} and Θ_{new}, based on the current solution, Ω_{old} and Θ_{old}, corresponding to the scout bee which found that site. For this, a neighborhood search strategy

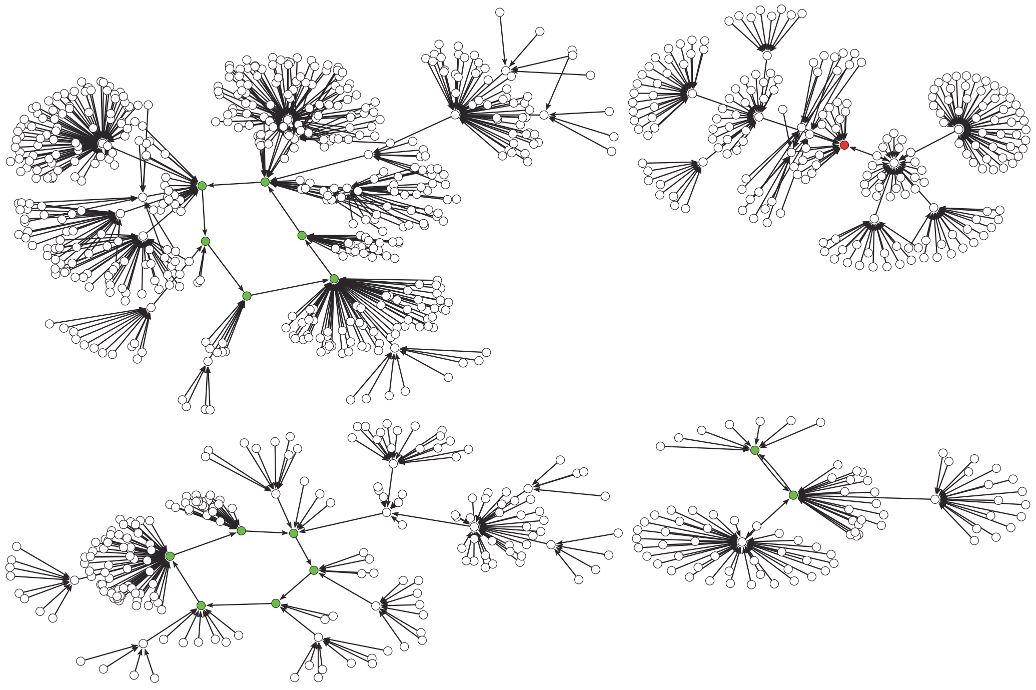

Fig. 8. State transition graph of the mammalian cell cycle network with the following updating scheme: (*CycD*, *p*27, *Cdc*20, *Cdh*1, *UbcH*10, *CycB*)(*E2F*)(*CycE*)(*Rb*, *CycA*). The red circle represents the fixed point state, the 14 green circles represent the states that belong to three limit cycles, two of length 6 and one of length 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

was used for the weight matrix (the pseudocode appears in Algorithm 4.2), and another for the threshold vector (the pseudocode appears in Algorithm 4.3).

Algorithm 4.2. Neighborhood search for the weight matrix

```
procedure NetSearch ($n$, $ngh$, $\Omega_{old}$)
  $\Omega_{new} \leftarrow \Omega_{old}$
  for $i = 1 : ngh$ do
    $x1 \leftarrow 1 + \text{round}((n-1) * \text{rand})$
    $x2 \leftarrow 1 + \text{round}((n-1) * \text{rand})$
    $x3 \leftarrow 1 + \text{round}((n-1) * \text{rand})$
    $x4 \leftarrow 1 + \text{round}((n-1) * \text{rand})$
    $y1 \leftarrow 1 + \text{round}((n-1) * \text{rand})$
    $y2 \leftarrow 1 + \text{round}((n-1) * \text{rand})$
    $y3 \leftarrow 1 + \text{round}((n-1) * \text{rand})$
    $y4 \leftarrow 1 + \text{round}((n-1) * \text{rand})$
    if rand $> 0.5$ then
      $\Omega_{new}(x1, y1) \leftarrow \Omega_{new}(x1, y1) + 1$
      if $\Omega_{new}(x1, y1) > 5$ then
        $\Omega_{new}(x1, y1) \leftarrow 5$
      end if
    else
      $\Omega_{new}(x1, y1) \leftarrow \Omega_{new}(x1, y1) - 1$
      if $\Omega_{new}(x1, y1) < -5$ then
        $\Omega_{new}(x1, y1) \leftarrow -5$
      end if
    end if
    $\Omega_{new}(x2, y2) \leftarrow 0$
    $\Omega_{new}(x3, y3) \leftarrow 0$
    $\Omega_{new}(x4, y4) \leftarrow 0$
  end for
  return $\Omega_{new}$
end procedure
```

Algorithm 4.3. Neighborhood search for the threshold vector

```
procedure VectorSearch ($n$, $\Theta_{old}$)
  $\Theta_{new} \leftarrow \Theta_{old}$
  for $i = 1 : n$
    if rand $\geq 0.5$ & $\Theta_{new}(i) < 2$ then
      $\Theta_{new}(i) \leftarrow \Theta_{new}(i) + 0.5$
    else if$\Theta_{new}(i) > -2$ then
      $\Theta_{new}(i) \leftarrow \Theta_{new}(i) - 0.5$
    else
      $\Theta_{new}(i) \leftarrow \Theta_{new}(i)$
    end if
  end for
  return $\Theta_{new}$
end procedure
```

Using the above definitions, the general bees algorithm is shown in Algorithm 4.4 .

Algorithm 4.4. The Bees Algorithm

```
procedure BA ($ns$, $ne$, $nb$, $nre$, $nrb$, $maxi$, $ngh$)
  Initialize a random population $\Omega_g$, for $g = 1, \ldots, ns$ and fixed values for $\Theta_g$
  Evaluate each candidate solution ($\Omega_g$, $\Theta_g$) in the fitness function
  $iter \leftarrow 0$
  while $iter < maxi$ do
    Search for $nre$ candidates in the neighborhood of the first $ne$ fittest solutions
    Search for $nrb$ candidates in the neighborhood of the $[ne + 1, ne + nb]$ fittest solutions
    Search for $ns - (ne + nb)$ candidates randomly
    Evaluate each new candidate solution in the fitness function
    ifthe fittest solution =0 then
      Break
    end if
    $iter \leftarrow iter + 1$
  end while
end procedure
```

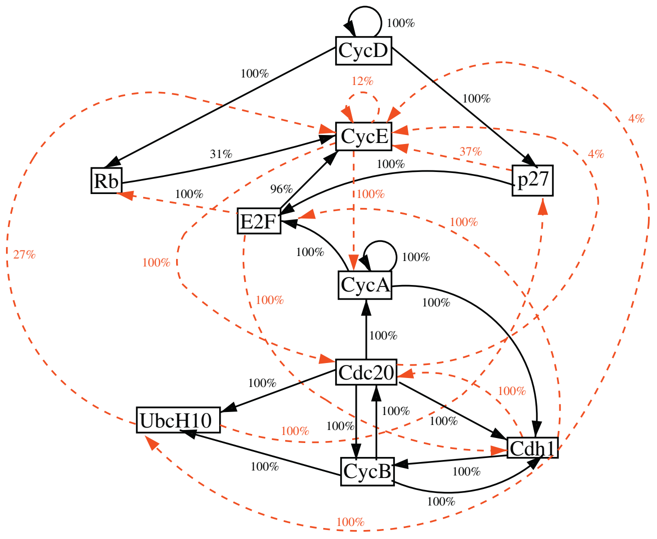

Fig. 9. Summary of the 100 synthetic networks. The percentage represents how many times the edge appeared. Edges that are also present in Fig. 2 are in black/solid and new edges in red/dashed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.4. Simulation setup

The BA algorithm was used to find 100 networks that contained exactly the two attractors of the original logical model. The following BA parameters were used $ns = 20$, $ne = 5$, $nb = 10$, $nre = 20$, $nrb = 5$, and $maxi = 4000$. For Algorithm 4.2, the neighborhood range parameter ngh was set to 3. These parameters were found empirically after several runs based on the effectiveness of the resulting networks.

4.5. Results

The 100 networks identified through this approach are summarized in Fig. 9. Each edge has a percentage that represents how many times that edge appeared within the 100 networks. Edges that were also present in Fig. 2 are shown in black/solid (color online) with new edges in red/dashed (color online). From Fig. 9 we can deduce which interactions in Fig. 2 are mandatory in order to obtain the same attractors. We notice that out of the 35 edges in Fig. 2, only 14 are present all the time in Fig. 9. There are 13 new iterations not present in the original model. *CycE*, which is a key regulator in the control of the G1-to-S transition in the cell cycle (Qu et al., 2003; Lilly and Spradling, 1996; Ables and Drummond-Barbosa, 2013), is the only node that shows more flexibility in the interaction, with incoming edges that are not present in all the 100 networks. It is interesting to see that *CycA*, which is known to promote S-phase entry in mammals (Chibazakura et al., 2011), can in fact be modeled by a threshold function and still obtain the same original attractors, this is accomplished by having as inputs *Cdc*20, *CycE*, and itself.

5. Conclusion

The mammalian cell cycle logical model and its equivalent threshold Boolean network model were analyzed. First we studied the behavior of the model under update perturbations, in particular, we analyzed what happens to the attractors of the network for all the possible deterministic dynamics. To do this, we separated the network's dynamics according to the two possible values of the *CycD* node, due to its constant nature. We have not only validated the results presented in Fauré et al. (2006) for the case *CycD* = 1; the existence of a single limit cycle of length seven and a fix point, under the parallel updating scheme, but we have also proved a simple result (Proposition 3.1) that explains this phenomenon whatever the updating scheme considered. Moreover, by using tools for the classification of different dynamics (Aracena et al., 2013), we have completely determined the full asymptotic behavior of the mammalian network under every deterministic updating mode for both cases; *CycD* = 1 and *CycD* = 0.

In the first case, we make some preliminary simplifications in the model which are equivalent to say that the subconfiguration (*CycD*, *Rb*, *p*27) = (1, 0, 0) is an *alliance* in the sense of Goles et al. (2013) because these values remain invariant against any deterministic update schedule. On the other hand, we observe that the range of the limit cycle lengths is between 2 and 7, being 4 the most frequent length (about 48% of all the different dynamics) and each of these cycles with a basin of attraction containing all the configurations such that *CycD* = 1. Furthermore, in the particular case of limit cycles of length 7, we exhibit all the possible limit cycles of the same length (evidently, in the presence of *CycD*) that can exist when the mammalian network is updated in a deterministic way. These limit cycles are not very different between them (Hamming distance bounded by 2), they have 3 common configurations and, additionally to the activations of nodes *Cdc*20 and *CycA* showed in Fauré et al. (2006), these limit cycles exhibit other simple transitions; the inhibition of nodes *UbcH*10, *Cdh*1 and *Cdc*20.

In the second case, it was observed that most of the dynamics have the fixed point, and only a few have additionally one or two limit cycles, with a maximum limit cycle length equal to 8, although the most frequent cycle length is 2. We notice that, although the mammalian cell cycle network model was constructed considering the parallel updating scheme, the limit cycles vary considerably in type and basin size, depending on the updating scheme employed, resulting in a model which is not very robust with respect to update perturbation. We could say that the limit cycle used to represent the cell cycle is an artifact caused by the parallel update in the sense of Klemm and Bornholdt (2005).

Given that genes are activated or inhibited in a fundamentally asynchronous way, GRN modelers, under the Boolean network approach, should try to associate cell differentiation or any biological behavior of interest to fixed points of the model, given that these are invariant to update changes (perturbations), therefore, eliminating the update issue when considering biological implementations.

To analyze the topology of the network, we adopted a reverse engineering approach, using BA to reconstruct threshold Boolean networks that contained exactly the same attractors as the original model. 100 networks were found, where it was possible to identify which interactions (edges) are present all the time, and new interactions that yield the same attractors. By relaxing the problem to having networks with the same attractors as the original logical model, it was found the *CycA* can be modeled by a threshold function, with inputs *Cdc*20, *CycE*, and itself.

Although the techniques used in this work were applied to analyze the mammalian cell cycle network, these are general, and could be employed to analyze other logical or threshold network models of gene regulatory networks.

Acknowledgments

The authors would like to thank Conicyt-Chile under grant Fondecyt 11110088 (G.A.R.), Fondecyt 1100003 (E.G.), Fondecyt 3130466 (M.M.), Basal(Conicyt)-CMM, and ANILLO ACT-88 for financially supporting this research.

References

Ables, E.T., Drummond-Barbosa, D., 2013. Cyclin E controls Drosophila female germline stem cell maintenance independently of its role in proliferation by modulating responsiveness to niche signals. Development 140, 530–540.

Akutsu, T., Miyano, S., Kuhara, S., 1999. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In: Pacific Symposium on Biocomputation, pp. 17–28.

Aracena, J., Demongeot, J., Fanchon, E., Montalva, M., 2013. On the number of different dynamics in Boolean networks with deterministic update schedules. Mathematical Biosciences 242, 188–194.

Aracena, J., Goles, E., Moreira, A., Salinas, L., 2009. On the robustness of update schedules in Boolean networks. Biosystems 97, 1–8.

Chibazakura, T., Kamachi, K., Ohara, M., Tane, S., Yoshikawa, H., Roberts, J.M., 2011. Cyclin A promotes S-phase entry via interaction with the replication licensing factor Mcm7. Molecular and Cellular Biology 31, 248–255.

Chowdhury, A.R., Chetty, M., Vinh, X.N., 2012. On the reconstruction of genetic network from partial microarray data, vol. 7663 LNCS of Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).

Davidich, M.I., Bornholdt, S., 2008. Boolean network model predicts cell cycle sequence of fission yeast. PLoS ONE 3 (2), e1672.

Demongeot, J., Elena, A., Sené, S., 2008. Robustness in regulatory networks: A multidisciplinary approach. Acta Biotheoretica 56, 27–49.

Fauré, A., Naldi, A., Chaouiya, C., Thieffry, D., 2006. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics 22, e124–e131.

Forghany, Z., Davarynejad, M., Snaar-Jagalska, B.E., 2012. Gene regulatory network model identification using artificial bee colony and swarm intelligence. In: 2012 IEEE Congress on Evolutionary Computation, CEC 2012.

Gérard, C., Goldbeter, A., 2009. Temporal self-organization of the cyclin/Cdk network driving the mammalian cell cycle. Proceedings of the National Academy of Sciences of the United States of America 106, 21643–21648.

Gérard, C., Goldbeter, A., 2011. A skeleton model for the network of cyclin-dependent kinases driving the mammalian cell cycle. Interface Focus 1, 24–35.

Goles, E., Montalva, M., Ruz, G.A., 2013. Deconstruction and dynamical robustness of regulatory networks: Application to the yeast cell cycle networks. Bulletin of Mathematical Biology 75, 939–966.

Gonzalez, O.R., Kuper, C., Jung, K., Naval Jr., P.C., Mendoza, E., 2007. Parameter estimation using simulated annealing for s-system models of biochemical networks. Bioinformatics 23, 480–486.

Kauffman, S., Peterson, C., Samuelsson, B., Troein, C., 2003. Random Boolean network models and the yeast transcriptional network. Proceedings of the National Academy of Sciences of the United States of America 100, 14796–14799.

Kauffman, S.A., 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22, 437–467.

Kentzoglanakis, K., Poole, M., 2012. A swarm intelligence framework for reconstructing gene networks: Searching for biologically plausible architectures. IEEE/ACM Transactions on Computational Biology and Bioinformatics 29, 358–371.

Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., Tomita, M., 2003. Dynamics modeling of genetic networks using genetic algorithm and s-system. Bioinformatics 19, 643–650.

Klemm, K., Bornholdt, S., 2005. Stable and unstable attractors in boolean networks. Physica Review E 72, 055101–055104.

Li, F., Long, T., Lu, Y., Ouyang, Q., Tang, C., 2004. The yeast cell-cycle network is robustly designed. Proceedings of the National Academy of Sciences of the United States of America 101, 4781–4786.

Liang, S., Fuhrman, S., Somogyi, R., 1998. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In: Pac. Symp. Biocomput, pp. 18–29.

Lilly, M.A., Spradling, A.C., 1996. The drosophila endocycle is controlled by Cyclin E and lacks a checkpoint ensuring S-phase completion. Genes and Development 10, 2514–2526.

Liu, G., Feng, W., Wang, H., Liu, L., Zhou, C., 2009. Reconstruction of gene regulatory networks based on two-stage Bayesian network structure learning algorithm. Journal of Bionic Engineering 6, 86–92.

Mendoza, L., Alvarez-Buylla, E.R., 1998. Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphogenesis. Journal of Theoretical Biology 193, 307–319.

Mendoza, M.R., Lopes, F.M., Bazzan, A.L.C., 2012. Reverse engineering of grns: An evolutionary approach based on the tsallis entropy. In: GECCO'12 – Proceedings of the 14th International Conference on Genetic and Evolutionary Computation, pp. 185–192.

Noman, N., Iba, H., 2007. Inferring gene regulatory networks using differential evolution with local search heuristics. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4, 634–647.

Novak, B., Tayson, J.J., 2004. A model for restriction point control of the mammalian cell cycle. Journal of Theoretical Biology 230, 563–579.

Pham, D.T., Castellani, M., 2009. The bees algorithm: modelling foraging behaviour to solve continuous optimization problems. Proceedings of IMechE Part C: Journal of Mechanical Engineering Science 223, 2919–2938.

Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M., 2006. The bees algorithm, a novel tool for complex optimisation problems. In: Proc. of the Second International Virtual Conference on Intelligent production machines and systems (IPROMS 2006), pp. 454–459.

Qu, Z., Weiss, J.N., MacLellan, W.R., 2003. Regulation of the mammalian cell cycle: a model of the G_1-to-S transition. American Journal of Physiology and Cell Physiology 284, C349–C364.

Repsilber, D., Liljenstrom, H., Andersson, S.G.E., 2002. Reverse engineering of regulatory networks: simulation studies on a genetic algorithm approach for ranking hypotheses. Biosystems 66, 31–41.
Ruz, G.A., Goles, E., 2010. Learning gene regulatory networks with predefined attractors for sequential updating schemes using simulated annealing. In: Proc. of IEEE the Ninth International Conference on Machine Learning and Applications (ICMLA 2010), pp. 889–894.
Ruz, G.A., Goles, E., 2012. Reconstruction and update robustness of the mammalian cell cycle network. 2012 IEEE Symposium on Computational Intelligence and Computational Biology, CIBCB 2012, 397–403.
Ruz, G.A., Goles, E., 2013. Learning gene regulatory networks using the bees algorithm. Neural Computing and Applications 22, 63–70.
Ruz, G.A., Timmermann, T., Goles, E., 2012. Building synthetic networks of the budding yeast cell-cycle using swarm intelligence. In: Proceedings – 2012 11th International Conference on Machine Learning and Applications, ICMLA 2012, pp. 120–125.
Wuensche, A., 2004. Basins of attraction in network dynamics: A conceptual framework for biomolecular networks. In: Modularity in Development and Evolution, pp. 288–311.
Xu, R., Wunsch II, D.C., Frank, R.L., 2007. Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4, 681–692.