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