Investigating the use of Boolean networks for the control of gene regulatory networks

Investigating the use of Boolean Networks for the Control of Gene Regulatory Networks

Nadia S. Taou, David W. Corne and Michael A. Lones

School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, UK

Abstract

The behaviour of biological cells emerges from complex patterns of interactions between genes and their products, known as gene regulatory networks (GRN). An important aim of biology is to control the dynamics of GRNs, in order to push a cell towards or away from certain behaviours. This could potentially be done by coupling a synthetic GRN to an existing biological GRN. In this work, we use Boolean networks, a methodology for modelling and simulating GRNs, to investigate the potential for doing this. Our results demonstrate that Boolean networks can be optimised to control other Boolean networks, and that the approach scales well as the target network size increases.

Keywords: Gene regulatory network, Boolean network, control system, evolutionary algorithm

2010 MSC: 00-01, 99-00 ***

1. Introduction

Complex networks are all around us: social networks, financial networks, ecological networks. Because of their ubiquity and their profound effects upon our lives and livelihood, there has been an increasing interest in controlling the dynamical behaviour of these networks [22]. However, complex networks have a range of properties that make them intrinsically difficult to control [43]. Much of the existing research in this area has focused on analytical methods, typically making use of conventional control theoretic approaches. This has resulted in

control methods that can be applied to certain types of complex network, for instance those with particular relationships between their structure and dynamics [22], and those with restricted topologies [11]. In this work, we take a more empirical approach, using evolutionary algorithms to explore the space of control interventions, and thereby automatically discovering interventions that are suitable for particular complex networks.

We consider the control of gene regulatory networks (GRNs) in particular. These are biochemical networks that involve genes and their protein products, especially the transcription factors that allow a gene to regulate another gene's expression. GRNs are fundamental to the behaviour of biological organisms, and control both the internal functions of individual biological cells and the overall development of multicellular organisms. In recent decades, there has been a concerted effort to characterise and map the GRNs of various organisms. However, there has been relatively little advancement in the control of GRNs. The benefits of controlling GRNs would be considerable, particularly from the perspective of medicine: for instance, being able to transition a cell from a cancerous state to a non-cancerous state, or being able to target the differentiation of a stem cell into a particular tissue type [32, 22].

In this paper, we focus on the use of Boolean networks [29] for modelling, simulating and controlling GRNs. A Boolean network is a considerably simplified model of a biological GRN. Nevertheless, there are numerous example of successfully using Boolean networks to capture the structure and dynamics of real biochemical networks [28, 4, 48, 17, 55, 16, 31, 47, 21, 44]. Boolean networks have also been adopted as a more general model of complex networks, and studies of their dynamical behaviour have given considerable insight into the properties of real world networks [5]. Perhaps lesser known, however, are the uses of Boolean networks (and computational models of GRNs more generally) within the computational intelligence and artificial life communities, where they are seen as a form of artificial intelligence somewhat akin to neural networks [39]. An example is the use of Boolean networks as robotic controllers [46], where they have shown the ability to generate robust control decisions through

the analysis of environmental inputs—which, in many respects, is comparable to the kind of control behaviour that biological GRNs carry out.

The work reported in this paper brings together the different roles of Boolean networks: using them both as a simulateable model of biological GRNs, and as a control system. More precisely, we use evolutionary algorithms to discover Boolean networks that can control the dynamics of other Boolean networks. We consider a number of things: the ability of evolutionary algorithms to optimise Boolean networks, the general ability of Boolean networks to control other Boolean networks, and the effect that topology has on both the difficulty of the control problem and the ability of the controllers. The eventual goal of this research is to design controllers for real biological GRNs, and to then implement these within biological cells. The use of Boolean networks is appropriate in this respect, since Boolean logic can be readily refined into biochemical implementations using synthetic biology principles [45].

This work builds upon initial results reported in [53]. In this paper, we report the results of a broader set of experiments using significantly larger Boolean networks and scale free Boolean networks. We also present first results showing that our method can control trajectories in a Boolean model of a real biological circuit. The paper is organised as follows: Section 2 reviews related research in the areas of Boolean modelling and control. Section 3 describes the methods used in this work, including a formal definition of Boolean networks and an overview of evolutionary algorithms. Section 4 presents results and analysis, and Section 5 concludes.

2. Related Literature

2.1. Boolean Modelling of Biological Networks

Modelling biological processes using quantitative and continuous mathematical models such as differential equations has brought important insights to systems biology [34, 1]. However, these models are often inefficient when simulating larger biological networks. This has promoted interest in discrete-valued mod-

els. One such model is the Boolean network, which models gene expression as a binary process (either on or off) and regulatory interactions as Boolean functions (e.g. AND, OR, NOT). The use of binary states and Boolean functions makes them especially cheap to simulate on a computer. Numerous studies have demonstrated that, despite their apparent simplicity and high level of biological abstraction, these models are often able to capture the qualitative dynamics of biological processes. For example, [28] developed a model of the yeast transcription network, [4] successfully modelled the GRN underlying pattern formation in drosophila, [16] modelled the quorum sensing circuits of *Pseudomonas aeruginosa* and [31] developed a model of the GPR142 biological pathway in type 2 diabetes. A number of studies have applied Boolean models to cancer analysis, both by considering specific pathways [47, 17, 21] and through more abstract systems-level studies [26, 27]. Many of these studies have carried out an attractor analysis of the resulting models in order to gain insights into the biological system's stable states [4, 26, 17], typically associating these with phenotypes. In [44], the authors went a stage further and identified nodes whose state would effect the accessible attractors, in effect identifying potential drug targets for preventing the expression of pathological phenotypes. Discrete models such as Boolean networks have been shown to be equivalent to continuous models when only the steady states of the system are considered [56]; however, it should be borne in mind that Boolean networks are not appropriate when a detailed quantitative understanding of a process is required. For a review of Boolean modelling in biology, see [47].

2.2. Controlling Boolean Networks

Finding strategies to control Boolean networks is an important and challenging problem. The control problem is typically defined in terms of leading a Boolean network's trajectory towards a particular point in its state space, ideally by manipulating the state of a minimum group of nodes and with the aim of reaching the target state in a minimal period of time. In common with the complex networks that they model, Boolean networks have a number of

characteristics that make them hard to control, including non-linear dissipative dynamics, multiple stable states and high dimensionality [43]. A number of previous works on the control of Boolean networks have been conducted [2, 11, 51, 33, 32, 43, 22]. Many of these use control theoretic approaches. For instance, pinning control methods have been used to stabilise the dynamics of Boolean networks, allowing particular phenotypic states to be maintained [35]. However, in general the control of Boolean networks is known to be NP-complete [2], meaning that optimal control techniques can only be applied to networks of limited size, though polynomial-time algorithms have been developed for Boolean networks with constrained topologies such as tree structures [2]. To an extent, the control problem can be made easier by identifying nodes that have dominant roles within the network (such as hubs in scale-free networks) and focusing control interventions on these [36, 32]. This works well for certain kinds of network, but in general it has been shown that dynamics can not be determined by structure alone, and therefore that methods based on structural analysis will not always be effective [22].

2.3. Computation and Control using Gene Regulatory Models

In addition to modelling biological GRNs, a number of studies have shown that GRN models can be used to carry out complex computational and control behaviours that are to some degree analogous to their biological activities (see [39] for a recent review). Typically this is done by training the model using a stochastic global optimiser such as an evolutionary algorithm (see section 7) so that it generates a particular behaviour. A number of approaches have used Boolean networks [19, 46, 24, 57], including [46], where a Boolean network was used for robotic control and [10] where asynchronous multiplexing behaviours were evolved. Examples using other GRN models include medical time series classification [40], chaos control [37], games controllers [15] and image compression [54]. A number of studies have shown that Boolean networks in particular have interesting computational characteristics [42, 24], and in situations where these models have been applied to control, the analysis of evolved controllers

typically shows a high degree of robustness in comparison to more conventional controller architectures [46, 15].

2.4. Implementing Boolean Networks in Cells

Part of the justification for using Boolean networks in this study is the potential for implementing them as optimised control systems within biological cells. One benefit of Boolean networks, in this respect, is that they are relatively amenable for implementation in biological cells using existing synthetic biology approaches. A key focus of synthetic biology has been on implementing digital circuits within cells, the idea being that this will allow more conventional computing approaches to be readily refined into biological systems. However, these approaches also have direct relevance to Boolean networks, since both digital circuits and Boolean networks are comprised of Boolean logic functions that can be implemented as logic gates. Synthetic biology has demonstrated that logic gates can be implemented in various biochemical forms, including proteins, RNA and DNA [45, 52, 50]. It is also possible to assemble these logic gates into circuits, though it remains challenging to implement large circuits due to crosstalk between logic gates [45]. Other authors have considered the potential for using synthetic biology to implement control systems: in [14], for example, the authors discuss how conventional control approaches may be refined into biological forms and used to control a cell's GRN.

Another benefit of using Boolean networks is that, because they are relatively abstract, they are less likely to be susceptible to the "reality gap" that is often found in computational modelling. This gap occurs when a model is optimised under simulation, but then does not function correctly when used in a real world setting. This is caused either by over-fitting to the simulation, or by noise in the real world system. Since Boolean networks have few parameters, they are less likely to over-fit than continuous-valued models of GRNs. Since relatively large signal differences are required to cause binary state changes, they are also likely to be less affected by noise. In this respect, the value of a Boolean approach has previously been demonstrated in the field of robotics, where Boolean network

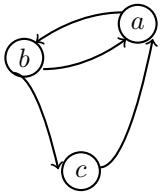


Figure 1: An example of a Boolean network with three nodes.

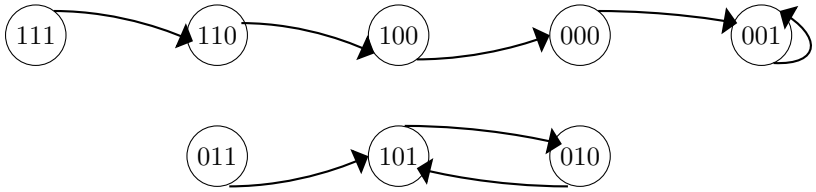


Figure 2: State transition graph corresponding to the Boolean network shown in Fig. 1.

abc (t)	abc (t+1)
000	001
001	001
010	101
011	101
100	000
101	010
110	100
111	110

(a) Truth table

$a^{(t+1)} = b^t$

$b^{(t+1)} = a^t \wedge c^t$

$c^{(t+1)} = \neg a^t$

(b) Boolean functions

Figure 3: Functions and truth table used by the Boolean network shown in Fig. 1.

controllers were found not to be susceptible to the reality gap [46].

3. Methods

3.1. Boolean Networks

We consider two forms of Boolean networks: Kauffman's original random Boolean network model [29], in which nodes have a uniform connectivity, and

scale-free Boolean networks, which capture the power law distribution of connectivity which is believed to occur widely in biological networks [7, 3, 6].

3.1.1. Random Boolean Networks

A Boolean network is a discrete-time non-linear dynamical system represented as a directed graph $G(N, E)$ composed of nodes, or vertices, V and edges E [29, 30]. A Random Boolean network (RBN) is a Boolean network which is randomly sampled from a set of possible BNs, i.e. the node inter-connections and the Boolean functions associated with each node are randomly generated [23, 18] (see Figs. 1–3). The time evolution of a RBN is expressed by a set of Boolean functions f_i, $i = 1, 2, 3, ...$ Each RBN node has a binary state s which is updated synchronously according to its Boolean function and the states of the k input nodes that are connected to it. Formally, $s(t+1) = f_i(s(t))$, where s is a set of network states $s \in \{0,1\}^N$, $t = 0, 1, 2, 3, 4, ...$ is the discrete time, $f_i : \{0,1\}^N \to \{0,1\}$. Since a RBN is deterministic $s(t+1)$ is only determined by $s(t)$. The possible number of Boolean functions is 2^{2^k}, and the state space is finite and equal to 2^N in size. Each node has $\frac{N!}{(N-k)!}$ possible ordered options for k different connections and the number of possible networks is $\left(\frac{2^{2^k}N!}{(N-k)!}\right)^N$. Since the state space is finite, states must eventually be repeated, leading to temporal structures called attractors. An attractor formed by one state is called a point attractor, and when it is formed by at least two states it is known as a cyclic attractor. Three dynamical regimes can be observed in RBNs: *ordered, chaotic* and *critical*. Ordered RBNs have attractors with a relatively short period, repeating the same series of states over and over again. Chaotic RBNs have long periods; although deterministic, they appear random. Critical RBNs also have attractors with long periods, but these appear to have a complex internal order which has been associated with computation. In general, the number of attractors grows with the number of nodes [29, 30, 9]. RBNs with $k < 2$ tend to be ordered; those with $k > 2$ tend to be chaotic; critical dynamics tend to be found when $k = 2$ [23].

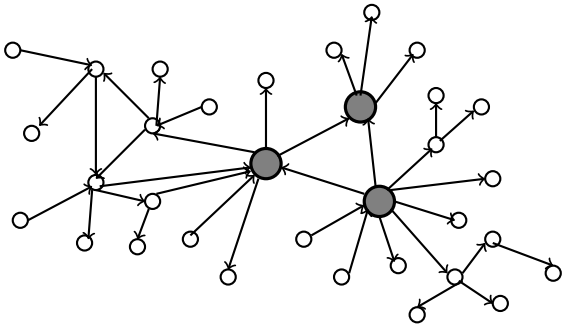


Figure 4: A scale free Boolean network, showing three hubs in grey.

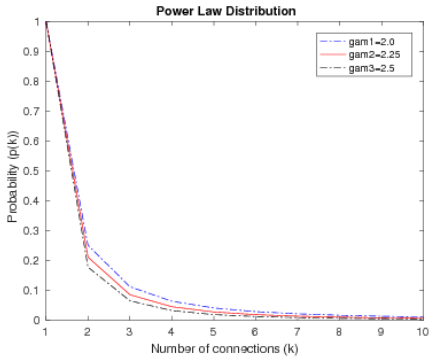


Figure 5: Power Law Distribution ($k = [1 - 10]$, $\gamma_1 = 2.0$ $\gamma_2 = 2.25$, $\gamma_1 = 2.5$). The plot shows the distribution of connectivities in scale free networks for three different values of γ. A large number of nodes have only a few connections, and a small number of nodes (hubs) have a large number of connections.

3.1.2. Scale Free Boolean Networks

RBNs typically have a fixed connectivity k. Real world complex networks, by comparison, tend to have a scale-free distribution of connectivities. A scale free Boolean network (SFBNs) is a Boolean network with a scale-free distribution, or more precisely a connected graph composed of a set of N nodes $\{N_1, N_2, ..., N_N\}$ and connectivities (k), or degree, which exhibits a power law distribution $P(k) \sim k^{\gamma}$ [5, 8, 12, 13, 49] (see Fig. 4). $P(k)$ is the probability distribution that an arbitrary node of the network is connected to n other nodes, and γ is the scale free exponent, or scaling parameter. Scale free ex-

ponents often lie in the range $2 < \gamma < 3$; however, there are some exceptions. SFBNs can be constructed by adding nodes incrementally to an existing network (*growth mechanism*) and by creating new connections to existing nodes with a *preferential attachment mechanism* i.e. new nodes will prefer to connect to more connected nodes. The probability p that a new node will be connected to a given node i depends on the number of existing connections of k_i that node i has. The mathematical expression of this probability is: $p \sim \frac{k_i}{\sum_d k_d}$, where k_i is the connectivity of node i and d is the index denoting the sum over network nodes. These two mechanisms explain the existence of *hubs*, which are nodes having connections with many other nodes in the network. Each node N_i has a binary state, either 0 or 1, and is connected to k_i other nodes of the network $\{N_{i1}, N_{i2}, ..., N_{ik_i}\}$ randomly chosen from a probability distribution $p_{inp}(k)$, $p_{inp}(k) = [(\sum_{n=1}^{\infty} k^{-\gamma})k^{\gamma}]^{-1}, \gamma > 1$. At each time step a Boolean function $f_i(N_{i1}, N_{i2}, ..., N_{ik_i})$ taken from a set of Boolean functions $F\{f_1, f_2, ..., f_N\}$ is assigned to N_i, such that for each state of k_i other nodes, $f_i = 1$ with probability p and $f_i = 0$ with probability $1 - p$. Each node of the network is updated synchronously as follows: $N_i(t+1) = f_i(N_{i1}(t), N_{i2}(t), ..., N_{ik_i}(t))$ and the entire network $\chi(t)$ is updated at time t with this dynamical equation: $\chi(t+1) = F(\chi(t))$, where $\chi(t) = \{N_1(t), N_2(t), ..., N_{(t)}\}$. SFBNs are more robust to external perturbations than Boolean networks [5]. In [5], Aldana showed that for most real scale free networks $\gamma \in [2, 2.5]$.

3.2. Evolutionary Algorithms

In this work, we use evolutionary algorithms (EAs) [20] to optimise both the structure and model parameters of Boolean networks. EAs are a form of evolutionary computation that mimic the natural process of Darwinian evolution in order to solve complex non-linear computational problems. In an EA, a population of candidate solutions is generated and iteratively evolved to search over the solution space of a problem (see Fig. 6). Each time a new solution is generated, it is evaluated using a fitness function that measures its objective value (or fitness). This value is then use to select between solutions in the

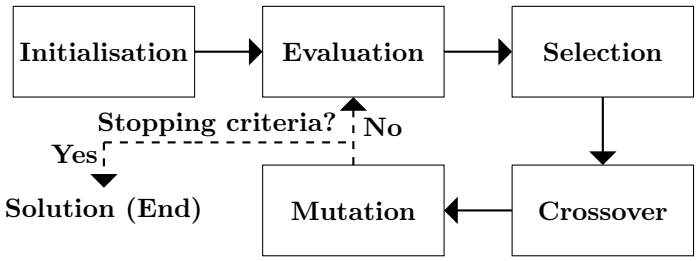


Figure 6: Evolutionary algorithm general framework

population, using the solutions with the best fitnesses to generate new solutions, whilst removing solutions with poor fitness from the population. New solutions are generated using mutation and recombination operators; mutation operators make small changes to existing solutions, recombination operators join parts of existing solutions to make new solutions. EAs are global optimisers (meaning they are relatively insensitive to local optima) and have much in common with other global optimisers, such as particle swarm optimisation [38]. There are various kinds of EA. In this work, we use an evolutionary algorithm with a vector-encoded solution; in this respect, it is very similar to a standard genetic algorithm (GA). It uses standard tournament selection, point mutation and uniform crossover. See Algorithm 1 for more details.

Algorithm 1 Training BN with GA

1: $P \leftarrow \{\}$
2: **for** *popsize* times **do** ▷ initialize population
3: $P \leftarrow P \cup \{\text{new random BN}\}$
4: **end for**
5: **for** each $p \in P$ **do** ▷ evaluate population
6: EVALUATE(p)
7: **end for**
8: **for** *maxgen* times **do**
9: $P' \leftarrow \{\}$
10: **while** $|P'| < |P|$ **do** ▷ create child solutions
11: $parent_1, parent_2 \leftarrow$ TOURNAMENTSELECT(p)
12: $child_1, child_2 \leftarrow$ RECOMBINE($parent_1, parent_2$)
13: $P' \leftarrow P' \cup$ MUTATE($child_1$) $\cup$ MUTATE($child_2$)
14: **end while**
15: **for** each $p \in P'$ **do** ▷ evaluate child population
16: EVALUATE(p)
17: **end for**
18: $P \leftarrow P'$ ▷ replace population with child population
19: **end for**
20: **return** P member with highest fitness

Algorithm 2 Evaluating a controller BN

$BN_d \leftarrow$ new random BN ▷ controlled BN
$S_d \leftarrow (0, 0, 0, \ldots), S_t \leftarrow (1, 1, 1, \ldots)$ ▷ initial and target states
$t \leftarrow 0$
while t within control period **do**
    $i \leftarrow 0$
    for each $c \in C_f$ **do** ▷ feedback from controlled BN to controller BN
        $s_{r_i} \leftarrow s_{d_c}, i \leftarrow i + 1$
    end for
    for t_r times **do** ▷ execute controller BN
        UPDATE(BN_r) ▷ apply each node's update function
    end for
    $i \leftarrow |BN_r|$
    for each $c \in C_I$ times **do** ▷ apply control interventions
        $s_{d_c} \leftarrow s_{r_i}, i \leftarrow i - 1$
    end for
    for t_d times **do** ▷ execute controlled BN
        UPDATE(BN_d)
        $t \leftarrow t + 1$
    end for
end while
$correct \leftarrow 0$
for each $s_{d_i} \in S_d, s_{t_i} \in S_t$ **do** ▷ compute distance from target state
    if $s_{d_i} = s_{t_i}$ **then**
        $correct \leftarrow correct + 1$
    end if
end for
$fitness \leftarrow \frac{correct}{|S_d|}$

3.3. Controller Evaluation

We use an EA to generate and optimise Boolean networks. The fitness function (see Algorithm 2) measures how well a Boolean network controls another Boolean network. We focus on the task of state space targeting, which means learning a control intervention that pushes a controlled Boolean network to a particular point in its state space. This problem is similar to the biological problem of controlling a GRN so that it moves to and remains within a particular region of its phenotype space.

All nodes in the controlled network have their expression state (S_d) set to 0 at the start of the control task, to maximise the initial distance from the target. For simplicity and clarity, the target state is all-1s, meaning that every node in the controlled network achieves a Boolean state of 1. This state is no easier or harder to reach than any other arbitrary state for a particular sample of controlled Boolean networks, and is not similar to the max-ones problem in the genetic algorithms literature. It is probable that, in practice, some controlled networks will be uncontrollable. Also it is plausible that the solution space will be hard to traverse for most controlled Boolean networks. For example, a solution which leads the controlled network to a state of all-but-one nodes turned on is unlikely to be proximal to a solution which leads the controlled network to the optimal state.

In order to avoid bias, we sample the controlled Boolean networks randomly. This means that, for many of the randomly sampled networks, it will not be possible to reach the optimum. Instead, it is intended that the fitness distribution over a number of runs will give a general insight into the ability of evolved Boolean network controllers to influence the dynamics of the controlled networks, and a measure of the degree to which they are able to achieve this. This is arguably more insightful than looking at their ability to control arbitrary Boolean networks derived from the biological literature, whose topologies and dynamics may not be representative of the wider class of GRNs.

We consider both RBNs and SFBNs for controller and controlled Boolean networks. All nodes in the controller network have their expression state (S_r)

randomly generated. For completeness, we look at each pairwise combination, i.e. RBNs controlling RBNs, RBNs controlling SFBNs, SFBNs controlling RBNs, and SFBNs controlling SFBNs. We consider different sizes of the controlled network in the range $[20, 50]$. For each combination of controller Boolean network type, controlled Boolean network type and controlled network size, we carry out 20 consecutive runs of the EA, each with a (very likely unique) randomly generated controlled network. For controlled SFBNs, scale free exponents in the range $\alpha \in [2, 2.5]$ are used.

To limit the combinatorial space of experiments, the controller size is fixed at 15 nodes. In previous work, we found the optimisation process to be relatively insensitive to controller size beyond a certain threshold [53]. This may reflect a trade-off between the greater computational resources available to larger controllers and the increased size of the search space that needs to be traversed in order to optimise them. However, it is also an indication that even relatively small Boolean networks are expressive, able to generate the dynamics necessary to solve the control task. This is fortunate, since large Boolean circuits remain challenging to implement using synthetic biology principles.

3.4. Controller Encoding and Evolutionary Parameters

An evolved controller is formed by a Boolean network (BN), a set of coupling terms, and two timing parameters. The controller BN is represented as an array of nodes, each comprising a Boolean function number (between 0 and 2^{2^k}), an initial state, and a set of input nodes, where each input is indicated by its position within the node array. It has been shown in [23] that an RBN's capacity for computation is maximal when it is in the critical regime; therefore, a value of $k = 2$ is used for RBNs, meaning that each node has precisely two inputs. The connectivity of each node in an SFBN is determined by sampling the power law distribution; for controllers, the number of connections for a particular node can change via mutation, so long as the power law distribution is maintained.

The coupling terms indicate the nodes in the controlled BN whose state will be changed by the controller BN (i.e. the control interventions, C_I) and the

nodes in the controlled BN whose state will be copied back to the controller BN (i.e. feedback connections, C_F), see Fig. 7. Inputs to the controller BN which are fed back from the controlled BN are always delivered by over-writing the states of nodes at the beginning of its node array. Control outputs are always read from the state of nodes at the end of the array. The number of coupling terms is uniformly sampled from and bounded to the range $[1, 5]$. The mutation operator can add, remove or modify coupling terms.

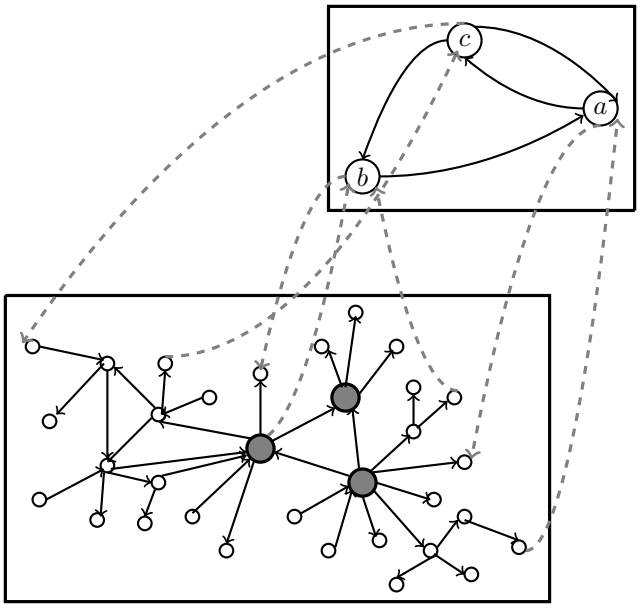
Two timing parameters are used when evaluating BNs. The first timing parameter determines how many steps (t_r) the controller BN will execute for each step (t_d) of the controlled BN, with values above 1 allowing the controller BN to execute faster than the controlled BN. The second parameter determines how often the controller BN is executed, in terms of the number of steps of the controlled BN. Larger values result in less frequent control interventions. Both timing parameters are uniformly sampled from and bounded to the range $[1, 100]$.

Fig. 8 shows an example of how a controller is linearly encoded. In each run, the evolutionary algorithm executed for 100 generations with a population size of 500, uniform crossover ($p = 0.15$) and point mutation ($p = 0.06$). These values were found to be appropriate during initial experiments. A controller's fitness is a measure of the distance between the controlled BN's final state and the target state, after a control period of 100 time steps of the controlled BN. This value is calculated by counting the number of 1s in the controlled BN's state at the final time step, and dividing by the size of the controlled BN, i.e. a value in the interval $[0, 1]$ where 1 indicates the correct all-ones state was reached.

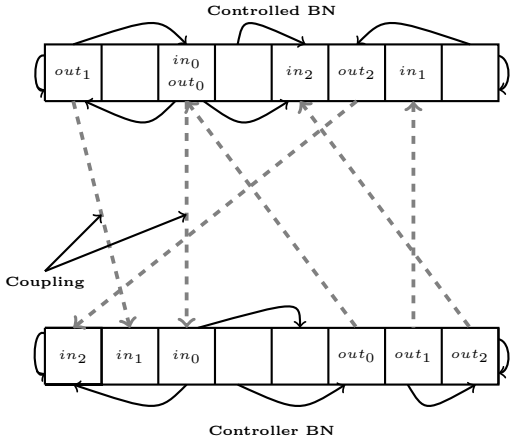
4. Results

4.1. Controlling randomly sampled Boolean networks

To provide a benchmark for optimisation, we first measured the natural dynamics of groups of randomly sampled RBNs and SFBNs, using the same



(a) Coupled Boolean network and scale free Boolean network.



(b) Linear encoding used by the evolutionary algorithm.

Figure 7: Boolean network coupled to a scale free Boolean network, also showing the linear encoding used by the evolutionary algorithm. Grey dashed arrows indicate coupling between controller BN and controlled BN.

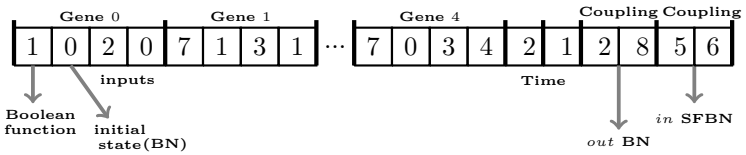
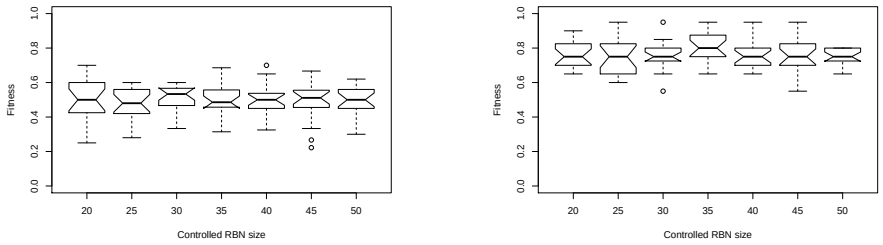


Figure 8: Example of a Boolean network's genetic representation. The timing and coupling terms indicate that this network is iterated twice each time it is executed, it is executed after every step of the controlled network, its control outputs (interventions) are copied to nodes 2 and 8 of the controlled BN, and its feedback (in) inputs from the controlled BN are copied from nodes 5 and 6.

fitness function used to evaluate evolved controllers. Figs. 9a and 10a show the fitness distributions in this case, i.e. indicating the fitnesses that can be achieved when no control is exerted over the target network. This shows that, without any control, Boolean networks tend towards a final state containing approximately equal numbers of 0s and 1s, indicated by a fitness of around 0.5 on average.

By comparison, Figs. 9b and 10b–10d show the fitness distributions achieved when controllers were evolved to perform state space targeting in randomly sampled SFBNs and RBNs. It is clear from these plots that fitness values are much higher on average when a controller is used, indicating that Boolean networks can be evolved to guide other Boolean networks towards particular parts of their state space. However, as expected, most runs do not find optimal controllers for the randomly sampled target networks.

It appears that, on average, there is no significant difference in the difficulty of the control problem regardless of whether the controlled networks are RBNs or SFBNs. Nevertheless, the fitness distributions for RBN targets are generally wider, indicating that there may be more instances that are hard to control. This is not unexpected, since the presence of hubs within SFBNs make it more likely that a signal injected into a randomly sampled network would percolate more widely, and therefore have an effect upon the network's dynamics. Figs. 10b–10d indicate that the choice of scale free exponent value had a relatively



(a) RBN natural dynamics

(b) RBN controlling RBN

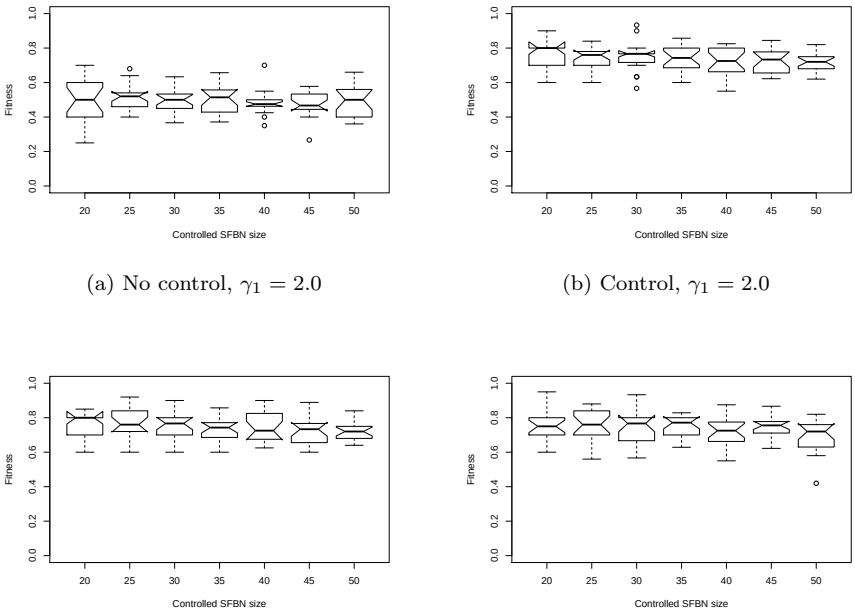
Figure 9: Fitness distributions for RBNs following their natural dynamics and under control. High fitness values are better. Notched box plots show summary statistics over 20 evolutionary runs. Overlapping notches indicate when median values (thick horizontal bars) are not significantly different at the 95% confidence level. Larger fitness values are better.

small impact on the difficulty of control, at least within the range we considered.

Although not a primary aim of this study, for completeness we also looked at using SFBNs to control other Boolean networks. Fig. 11 summarises these experiments. Surprisingly, these show that it is significantly harder to evolve SFBNs to carry out control in comparison to RBNs, regardless of whether the target networks are RBNs or SFBNs. This is rather contrary to previous assertions that SFBNs are more evolvable than networks with uniform connectivity [5]. More work is required to understand why this is the case, though it is possible that it is a problem specific observation, or that it is related to the choice of evolutionary parameters used in this study. However, it does suggest that topology is an important consideration when optimising Boolean networks to carry out control.

4.2. Controlling a Boolean model of a biological network

The results presented above give a clear indication that it is possible to evolve RBNs that can influence the dynamics of other BNs by controlling their trajectory towards an arbitrary target state. However, in real biological control problems, the target state is determined by the biological context and is often a



(a) No control, $\gamma_1 = 2.0$

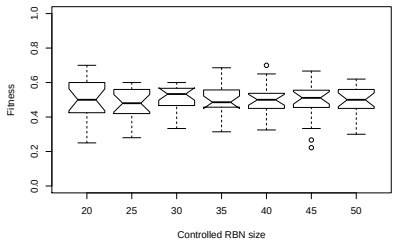
(b) Control, $\gamma_1 = 2.0$

(c) Control, $\gamma_2 = 2.25$

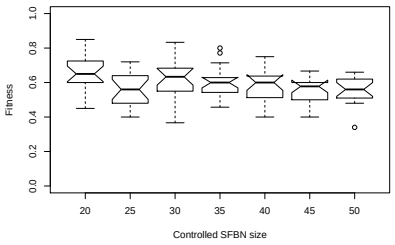
(d) Control, $\gamma_3 = 2.5$

Figure 10: Fitness distribution of SFBNs (a) following their natural dynamics, and (b–d) controlled with evolved RBNs, $\gamma_1 = 2.0$, $\gamma_2 = 2.25$ and $\gamma_3 = 2.5$.

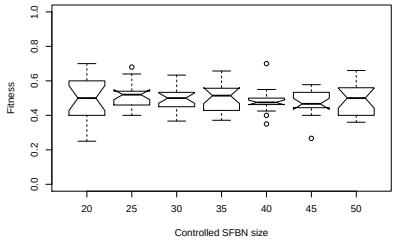
stable state of the system. To give an indication of how well our approach works when applied to these kind of problems, we evolved RBNs to control trajectories in a well known Boolean model of the regulatory system governing T helper cell differentiation [41]. This system has 23 nodes and is known to have three stable states, or point attractors, which correspond to different cell types. The control task (which we adapted from [25]) is to guide a trajectory to a specified stable state, beginning from an initial, randomly sampled, state. This is repeated for each of the three stable states, carrying out 20 runs of the evolutionary algorithm for each target state, with each run having a different initial state. The same fitness function is used, i.e. the distance from the target state at the end of the control period. To give an indication of the difficulty of the task, we also repeated these runs without control, showing the distribution of distances



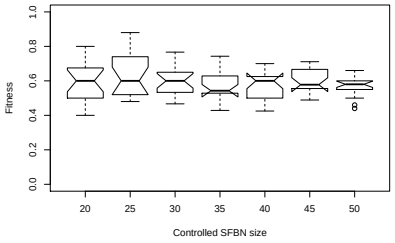
(a) RBN natural dynamics



(b) SFBN controlling RBN



(c) SFBN natural dynamics, $\gamma_2 = 2.25$



(d) SFBN controlling SFBN, $\gamma_2 = 2.25$

Figure 11: Fitness distributions for SFBNs evolved to control RBNs and SFBNs.

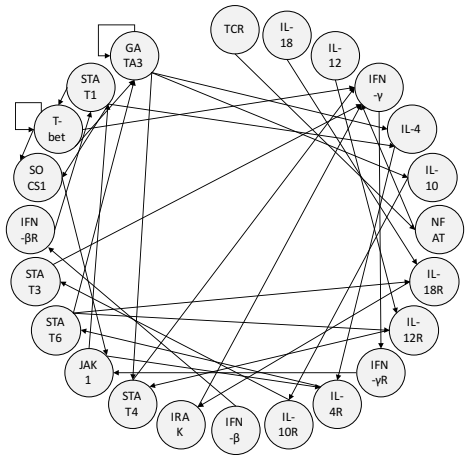
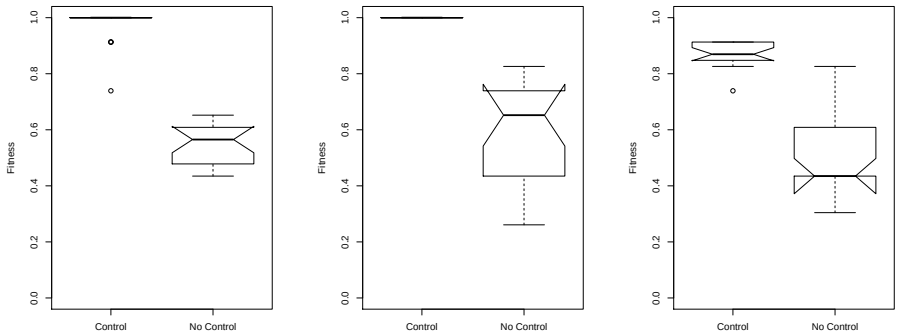


Figure 12: The T helper cell differentiation regulatory system, showing the interactions between the 23 nodes of the Boolean model. See [41] for details of regulatory functions.



(a) Stable state 1 (b) Stable state 2 (c) Stable state 3

Figure 13: Fitness distributions for the T helper cell differentiation control problem, showing the normalised distances from the target for each of the system's stable states both with and without control when starting at randomly sampled initial states.

from the target attractor when the system follows its natural dynamics from a set of randomly sampled initial points. As before, a controller RBN size of 15 was used.

Fig. 13 shows the results. It can be seen that the EA successfully finds RBNs that can control the dynamics of the system for two of the three stable states. For these two control problems, optimal controllers are found in almost all of the runs. Whilst the EA did not find optimal controllers for the third stable state, it did get close, with the highest fitness final states differing from the target by only two nodes. This may indicate a deceptive local optimum in the search space, from where it is difficult to transition to the target state. If this is the case, then it might be possible to improve the results by using a diversity preserving mechanism. Nevertheless, it is notable that none of the stable states were reached by the system when it was following its natural dynamics, indicating that these are not trivial control problems.

5. Conclusions

Appropriate and efficient control of gene regulatory networks is required in order to change the behaviour of biological cells. However, previous work in this field indicates that this control problem is very hard, and can only be solved analytically when network topologies have a number of restrictions. In this paper we have shown that Boolean networks, a class of computational models of gene regulatory networks, can be optimised to control the trajectories of other Boolean networks using an evolutionary algorithm. We presented results using both randomly sampled Boolean networks (with uniform and scale-free topologies) and a Boolean model of a real biological regulatory circuit. Significantly, our results demonstrate relatively good scalability as the size of the target network increases. This is in contrast to previous approaches using optimal control techniques, where scalability has been seen as a major limitation.

Our method has several novel elements, including the use of a generative model to produce control sequences, and the use of search-based optimisation techniques to find control interventions that are suitable for particular target networks. To our knowledge, this is the first demonstration of the use of a closed-loop controller to govern state space targeting within a cell's regulatory network. This is significant because, in principle, Boolean models could be refined into synthetic biology circuits, providing a potential route to *in vivo* implementation of a controller. Assuming the resulting circuit could be delivered to the target cell or tissue, this would enable direct control of disease processes caused by biological gene regulatory network entering an undesired state, using the synthetic circuit to guide the network's trajectory back to a healthy state. The use of artificial gene regulatory networks to do this also allows us to leverage the observed complexity, robustness and efficiency of these natural control systems.

In future work, we aim to look more closely at the issues involved with controlling real biological regulatory networks, for instance dealing with stochasticity and different time sales. We will also look more closely at how the optimised

controllers achieve control, and use this knowledge to improve wider understanding of how to control complex dynamical networks.

References

[1] Tatsuya Akutsu. Mathematical models and computational methods for inference of genetic networks. In Hitoshi Iba and Nasimul Noman, editors, *Evolutionary Computation in Gene Regulatory Network Research*, pages 30–48. John Wiley & Sons, 2016.

[2] Tatsuya Akutsu, Morihiro Hayashida, Wai-Ki Ching, and Michael K Ng. Control of boolean networks: hardness results and algorithms for tree structured networks. *Journal of Theoretical Biology*, 244(4):670–679, 2007.

[3] Reka Albert. Scale-free networks in cell biology. *Journal of cell science*, 118(21):4947–4957, 2005.

[4] Réka Albert and Hans G Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *Journal of theoretical biology*, 223(1):1–18, 2003.

[5] Maximino Aldana. Boolean dynamics of networks with scale-free topology. *Physica D: Nonlinear Phenomena*, 185(1):45–66, 2003.

[6] Eivind Almaas and Albert-László Barabási. Power laws in biological networks. In *Power laws, scale-free networks and genome biology*, pages 1–11. 2006.

[7] M Madan Babu, Nicholas M Luscombe, L Aravind, Mark Gerstein, and Sarah A Teichmann. Structure and evolution of transcriptional regulatory networks. *Current opinion in structural biology*, 14(3):283–291, 2004.

[8] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific American*, 288(5):50–59, 2003.

[9] Sven Bilke and Fredrik Sjunnesson. Stability of the kauffman model. *Physical Review E*, 65(1):016129, 2001.

[10] Larry Bull and Richard Preen. On dynamical genetic programming: Random boolean networks in learning classifier systems. In *Genetic Programming, Proceedings of the 12th European Conference on Genetic Programming, EuroGP 2009, Leonardo Vanneschi, Steven Gustafson, Alberto Moraglio, Ivanoe De Falco and Marc Ebner, 5481, LNCS*, pages 37–48. Springer, 2009.

[11] Daizhan Cheng and Hongsheng Qi. Controllability and observability of boolean control networks. *Automatica*, 45(7):1659–1667, 2009.

[12] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[13] Reuven Cohen, Shlomo Havlin, and Daniel Ben-Avraham. Structural properties of scale free networks. *Handbook of graphs and networks*, 2003.

[14] José ER Cury and Fabio L Baldissera. Systems biology, synthetic biology and control theory: a promising golden braid. *Annual Reviews in Control*, 37(1):57–67, 2013.

[15] Sylvain Cussat-Blanc, Jean Disset, Stephane Sanchez, and Yves Duthen. Artificial gene regulatory networks for agent control. *Evolutionary Computation in Gene Regulatory Network Research*, page 301, 2016.

[16] Stylianos E Dallidis and Ioannis G Karafyllidis. Boolean network model of the pseudomonas aeruginosa quorum sensing circuits. *IEEE transactions on nanobioscience*, 13(3):343–349, 2014.

[17] Maria I Davidich and Stefan Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PloS one*, 3(2):e1672, 2008.

[18] Barbara Drossel. Random boolean networks. *Reviews of nonlinear dynamics and complexity*, 1:69–110, 2008.

[19] Elena Dubrova, Maxim Teslenko, and Hannu Tenhunen. A computational scheme based on random boolean networks. In *Transactions on Computational Systems Biology X*, pages 41–58. Springer, Berlin Heidelberg, 2008.

[20] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2008.

[21] Herman F Fumiã and Marcelo L Martins. Boolean network model for cancer pathways: predicting carcinogenesis and targeted therapy outcomes. *PloS one*, 8(7):e69008, 2013.

[22] Alexander J Gates and Luis M Rocha. Control of complex networks requires both structure and dynamics. *Scientific reports*, 6, 2016.

[23] Carlos Gershenson. Introduction to random boolean networks. *In Bedau, M., P. Husbands, T. Hutton, S. Kumar, and H. Suzuki (eds.) Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, pages 160–173, 2004.

[24] Alireza Goudarzi, Christof Teuscher, Natali Gulbahce, and Thimo Rohlf. Emergent criticality through adaptive information processing in boolean networks. *Physical review letters*, 108(12):128702, 2012.

[25] Wenpin Hou, Takeyuki Tamura, Wai-Ki Ching, and Tatsuya Akutsu. Finding and analyzing the minimum set of driver nodes in control of boolean networks. *Advances in Complex Systems*, 19(03):1650006, 2016.

[26] Sui Huang, Gabriel Eichler, Yaneer Bar-Yam, and Donald E Ingber. Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Physical review letters*, 94(12):128701, 2005.

[27] Sui Huang, Ingemar Ernberg, and Stuart Kauffman. Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. In *Seminars in cell & developmental biology*, volume 20, pages 869–876. Elsevier, 2009.

[28] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein. Random boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences*, 100(25):14796–14799, 2003.

[29] Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[30] Stuart A Kauffman. *The origins of order: Self organization and selection in evolution*. 1993.

[31] Aman Chandra Kaushik and Shakti Sahi. Boolean network model for gpr142 against type 2 diabetes and relative dynamic change ratio analysis using systems and biological circuits approach. *Systems and synthetic biology*, 9(1-2):45–54, 2015.

[32] Junil Kim, Sang-Min Park, and Kwang-Hyun Cho. Discovery of a kernel for controlling biomolecular regulatory networks. *Scientific reports*, 3, 2013.

[33] Kaoru Kobayashi and Kunihiko Hiraishi. A petri net-based approach to control of boolean networks. In *Networking and Computing (ICNC), 2012 Third International Conference on*, pages 399–403, 2012.

[34] Nicolas Le Novère. Quantitative and logic modelling of molecular and gene networks. *Nature Reviews Genetics*, 16(3):146–158, 2015.

[35] Fangfei Li. Pinning control design for the stabilization of boolean networks. *IEEE Transactions on Neural Networks and Learning Systems*, 27(7):1585–1590, 2016.

[36] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

[37] Michael Lones, Alexander P Turner, Leo SD Caves, Susan Stepney, Stephen L Smith, and Andy M Tyrrell. Artificial biochemical networks:

Evolving dynamical systems to control dynamical systems. *Evolutionary Computation, IEEE Transactions on*, 18(2):145–166, 2014.

[38] Michael A Lones. Metaheuristics in nature-inspired algorithms. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 1419–1422. ACM, 2014.

[39] Michael A. Lones. Computing with artificial gene regulatory networks. *H. Iba and N. Noman (eds), Evolutionary Algorithms in Gene Regulatory Network Research, Wiley*, 2016.

[40] Michael A Lones, Stephen L Smith, Andy M Tyrrell, Jane E Alty, and DR Stuart Jamieson. Characterising neurological time series data using biologically motivated networks of coupled discrete maps. *BioSystems*, 112 (2):94–101, 2013.

[41] Luis Mendoza and Ioannis Xenarios. A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theoretical Biology and Medical Modelling*, 3(1):13, 2006.

[42] Bertrand Mesot and Christof Teuscher. Deducing local rules for solving global tasks with random boolean networks. *Physica D: Nonlinear Phenomena*, 211(1):88–106, 2005.

[43] Adilson E Motter. Networkcontrology. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(9):097621, 2015.

[44] Arnaud Poret and Jean-Pierre Boissel. An in silico target identification using boolean network attractors: avoiding pathological phenotypes. *Comptes rendus biologies*, 337(12):661–678, 2014.

[45] Oliver Purcell and Timothy K Lu. Synthetic analog and digital circuits for cellular computation and memory. *Current opinion in biotechnology*, 29: 146–155, 2014.

[46] Andrea Roli, Mattia Manfroni, Carlo Pinciroli, and Mauro Birattari. On the design of boolean network robots. In *Applications of Evolutionary Computation, Proceedings of the 2011 International Conference on Applications of Evolutionary Computation - Volume Part I, EvoApplications'11*, pages 43–52. Springer, Berlin Heidelberg, 2011.

[47] Assieh Saadatpour and Réka Albert. Boolean modeling of biological regulatory networks: a methodology tutorial. *Methods*, 62(1):3–12, 2013.

[48] Julio Saez-Rodriguez, Luca Simeoni, Jonathan A Lindquist, Rebecca Hemenway, Ursula Bommhardt, Boerge Arndt, Utz-Uwe Haus, Robert Weismantel, Ernst D Gilles, Steffen Klamt, et al. A logical model provides insights into t cell receptor signaling. *PLoS Comput Biol*, 3(8):e163, 2007.

[49] Roberto Serra, Marco Villani, and Luca Agostini. On the dynamics of scale-free boolean networks. In *Neural Nets*, pages 43–49. 2003.

[50] Xiaolong Shi, Zhiyu Wang, Chenyan Deng, Tao Song, Linqiang Pan, and Zhihua Chen. A novel bio-sensor based on dna strand displacement. *PloS one*, 9(10):e108856, 2014.

[51] Chen Shi-Jian and Hong Yi-Guang. Control of random boolean networks via average sensitivity of boolean functions. *Chinese Physics B*, 20(3): 036401, 2011.

[52] Vijai Singh. Recent advances and opportunities in synthetic logic gates engineering in living cells. *Systems and synthetic biology*, 8(4):271–282, 2014.

[53] Nadia S Taou, David W Corne, and Michael A Lones. Towards intelligent biological control: Controlling boolean networks with boolean networks. In *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016*, volume 9597 of *Lecture Notes in Computer Science*, pages 351–362, 2016.

[54] Martin A. Trefzer, Tuze Kuyucu, Julian F. Miller, and Andy M. Tyrrell. Image compression of natural images using artificial gene regulatory networks. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 595–602, New York, NY, USA, 2010. ACM.

[55] Alan Veliz-Cuba and Brandilyn Stigler. Boolean models can explain bistability in the lac operon. *Journal of computational biology*, 18(6):783–794, 2011.

[56] Alan Veliz-Cuba, Joseph Arthur, Laura Hochstetler, Victoria Klomps, and Erikka Korpi. On the relationship of steady states of continuous and discrete models arising from biology. *Bulletin of mathematical biology*, 74(12): 2779–2792, 2012.

[57] Massimiliano Zanin and Alexander N Pisarchik. Boolean networks for cryptography and secure communication. *Nonlinear Science Letters B: Chaos, Fractal and Synchronization. Vol*, 1(1):27–34, 2011.