BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 16 2010, pages 2037–2041
doi:10.1093/bioinformatics/btq333

Systems biology

Advance Access publication July 2, 2010

Modeling and analyzing complex biological networks incooperating experimental information on both network topology and stable states

Yi Ming Zou*

Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, USA

Associate Editor: Trey Ideker

ABSTRACT

Motivation: Linking the topology of a complex network to its long-term behavior is a basic problem in network theory, which has been on the focus of many recent research publications. To obtain a suitable Boolean model for a biological system, one must analyze the initial model and compare it with other experimental evidence, and if necessary, make adjustments by changing the topology of the wiring diagram. However, our knowledge on how to link the topology of a network to its long-term behavior is very limited due to the complexity of the problem. Since the need to consider complex biological networks has become ever greater, develop both theoretical foundation and algorithms for model selection and analysis has been brought to the forefront of biological network study.

Results: This article proposes a novel method to study intrinsically the relationship between experimental data and the possible Boolean networks, which can be used to model the underlying system. Simple and easy to use criteria for a Boolean network to have both a given network topology and a given set of stable states are derived. These criteria can be used to guide the selection of a Boolean network model for the system, as well as to gain information on the intrinsic properties, such as the robustness and the evolvability, of the system. A Boolean model for the fruit fly Drosophila melanogaster is used to explain the method.

Contact: ymzou@uwm.edu

Received on March 16, 2010; revised on June 12, 2010; accepted on June 15, 2010

1 INTRODUCTION

Boolean networks were introduced as random models of genetic regulatory networks to study biological systems by Kauffman (1969), and in recent years, we have seen a rapid increase of applications of Boolean networks to modeling and analyzing biological systems, as well as a rapid increase of research activities aim at developing foundational theories and algorithms for Boolean networks to address questions arise from biological applications. See for example, Bonneau (2008), Davidson and Levine (2008), Karlebach and Shamir (2008), Kinoshita *et al.* (2009), Purnick and Weiss (2009), Tomlin and Axelrod (2007) and the references therein.

The construction of a Boolean model for a biological system, such as a gene regulatory network, is based on the experimental evidence on the interactions among the genes (Albert and Othmer, 2003; Aldana and Cluzel, 2003; Feist *et al*., 2009; Karlebach and Shamir, 2008; Kauffman, 1969). The data can be integrated into a diagram, which is similar to an electrical wiring diagram (Bennett and Hasty, 2008; Hasty *et al*., 2002), and logical functions that define a Boolean network for the biological system can then be formulated according to the topology (how it is wired) of the diagram (Fig. 1).

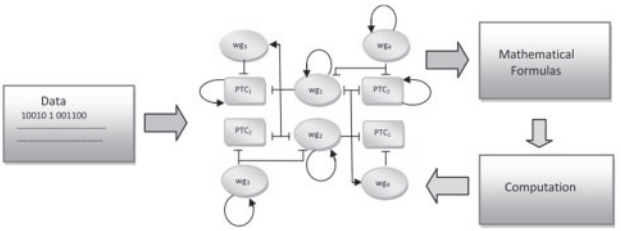

Fig. 1. The middle block depicts a simplified Boolean network, which was formulated to model the expression pattern of the segment polarity genes in the fruit fly *Drosophila melanogaster* (Albert and Othmer, 2003). The interactions among the genes are specified by the lines. Arrows mean activations and blunts mean inhibitions. To distinguish the different actions of the genes among themselves and to make it clear from the diagram which of the logical operations $\vee$, $\wedge$ and $\neg$ should be used in formulating the updating functions, two copies of each of the genes wg_3, wg_4, PTC_1 and PTC_2 were drawn, with the assumption that the identical copies were to be combined by the logical operation $\vee$. More details are provided in the Supplementary Material.

For a system that only partial information was obtained through experiments, the choice of a Boolean network for the system is non-unique, assumptions must be made in order to construct a wiring diagram, and linking the topology of the wiring diagram to the stable states of the network constructed from it becomes crucial (Pal *et al*., 2005; Pomerance *et al*., 2009; Strogatz, 2001). If the system under consideration is small, then we can use the exhaustive method to study the system effectively. For systems occurring in biology, which are typically large, developing general theory to guide our study is important. However, for complex systems, only in some special cases, we have a clear picture of how to derive information on the long-term behavior of a network from the topology of the corresponding wiring diagram.

Here, we briefly review some recent publications that are relevant to this study. Colón-Reyes *et al.* (2004) studied a family of Boolean networks for which each updating function (see Section 2 for definition) is given by a *monomial*. A method to derive the dynamical properties of such a Boolean network from the structure of the monomials that define the network was derived there, and thus

*To whom correspondence should be addressed.

provided a link from the network topology to its long-term behavior. Kauffman *et al.* (2004) found that a random Boolean genetic network model with canalyzing rules is dynamically stable. Nested canalyzing Boolean networks were further studied in Jarrah *et al.* (2007b). Nochomovitz and Li (2006) explored the mapping from the space of network topologies to the space of dynamical phenotypes for three- and four-node networks using the exhaustive method, and demonstrated that certain dynamical phenotypes can be generated by an atypically broad spectrum of network topologies. A review of the aforementioned and other existing literature shows that as far as linking the dynamics of a complex network to its topology and providing efficient algorithm (polynomial time in the number of nodes) to derive such information are concerned, there has yet to be substantial progress beyond linear systems and monomial systems. In fact, the general problem is believed to be non-deterministic polynomial-time (NP) hard (Tamura and Akutsu, 2009; Zhang *et al.*, 2007).

Since a convenient way to handle a Boolean network is to use its polynomial representation, it is natural to employ algebraic method to study Boolean networks. Attempts had been made to utilize the existing algebraic geometry tools to develop algorithms for handling the computational needs in modeling biological systems with Boolean networks. Laubenbacher and Stigler (2004) and Jarrah *et al.* (2007a) treated finite dynamical systems on a more general setting (which includes Boolean networks as a special case) using algebraic geometry. As of the writing of this article, existing algorithms and softwares from computational algebraic geometry can only handle relatively small size systems except Boolean networks, for which Brickenstein and Dreyer (2009) has developed an algorithm that specifically aims at the computations for Boolean polynomials. According to their statements, their software can handle industrial size datasets.

There are also other recent studies devoted to deriving information on the long-term behavior of a biological system from the structure of a Boolean model. Choudhary *et al.* (2006) derived a control algorithm by minimizing a composite finite-horizon cost function that is a weighted average over the individual networks of a family of Boolean networks possessing a common set of stable states (see Section 2 for definition). Xiao and Dougherty (2007) studied the impact of function perturbations in Boolean networks by focusing on function perturbation in the form of a one-bit change of the truth table and investigated its impact on the stable states. These studies can be viewed as attempts to link the topology of a Boolean network to its set of stable states.

In this article, we address the following question: giving partial information on a wiring diagram and a set of states, can we formulate easy to verify conditions (criteria) that will ensure a Boolean network we construct is consistent with the given information on the wiring diagram as well as capable of producing the given set of states as its stable states?

We remark that the problem of generating Boolean networks with given attractor structure has been studied before. See for example, Pal *et al.* (2005) and the references therein. In particular, Pal *et al.* (2005) provided an analysis of the complexity of the problem, and developed two algorithms for generating Boolean networks with prescribed attractor structures based on the *truth table*. However, the problem of connecting both information on the topology of the wiring diagram of a Boolean network and the set of stable states were not discussed. The algorithms developed in Laubenbacher and Stigler (2004) and Jarrah *et al.* (2007a) also produce Boolean networks that satisfy the property that each of the given stable states transmitted to itself. The main difference between the problem considered there and the problem considered here is that the problem considered here is about the connection between the given information on a wiring diagram and the given information on stable states, so the consistency between these two pieces of information must be considered when a Boolean network is constructed; while their algorithms start with a given set of transitions and produce a dependency diagram. Though a given set of transitions carries partial information on a wiring diagram, the information is more local in nature compare with the information carried by the given conditions on the wiring diagram. For instance, to give the state transitions for the fact that a node is activated by another node requires using half of the total states in the state space.

In terms of a truth table, the problem we want to address is: given a partially filled truth table as well as partial information on the interactions among the genes considered, how can we complete the table so the resulting Boolean network is both consistent with the information on the known interactions and capable of producing the given set of states as its stable states? We develop a new approach for this problem here. Our approach leads to some easy to apply criteria for the existence of Boolean networks, which fit the experimental evidence on gene interactions and are capable of producing the observed stable states. In a sense, we provide a link from a wiring diagram to the stable states of the Boolean network constructed from it. We also describe a method for constructing these Boolean networks if the criteria are met. Our theoretical results can be used to derive information on the intrinsic properties of Boolean networks and are suitable for computer implement. We expect our results to be useful in real-world applications, such as guiding the selection of a Boolean model and the construction of synthetic gene circuits (Purnick and Weiss, 2009; Tyers and Mann, 2003).

2 MODEL

Boolean networks can be conveniently represented using Boolean value polynomials, which are suitable for both analysis and computation. If a Boolean network has n nodes, then we can use n Boolean-valued polynomials $f_1,\ldots,f_n$ in n variables $x_1,\ldots,x_n$ to represent it by labeling the nodes as $1,2,\ldots,n$. The variables $x_1,\ldots,x_n$ take the values 0 and 1, and the polynomials $f_1,\ldots,f_n$ also take the values 0 and 1. We have the options to either use the logical operations **OR** ($\vee$), **AND** ($\wedge$) and **NOT** ($\neg$), or the modulo 2 arithmetic operations addition and multiplication to perform the calculations for Boolean variables and polynomials. The correspondence is as follows:

$$x_i \wedge x_j = x_i x_j,\ x_i \vee x_j = x_i + x_j + x_i x_j,\ \neg x_i = x_i + 1.$$

When studying the dynamical properties of a Boolean network, we consider the time-discrete dynamical system defined by these Boolean functions on the state space $\{0,1\}^n$ (all the sequences of length n formed by 0 and 1)

$$\mathbf{f} = (f_1,\ldots,f_n): \{0,1\}^n \to \{0,1\}^n, \tag{1}$$
$$\mathbf{f}: (x_1(t),\ldots,x_n(t)) \mapsto (x_1(t+1),\ldots,x_n(t+1)).$$

That is, the functions f_i, $1 \le i \le n$, give the updating rules for the nodes, and the state of the i-th node at time $t+1$ is given by the

function value $f_i(x_1(t),\ldots,x_n(t))$. For gene regulatory networks, the variables $x_1,\ldots,x_n$ represent the genes and the functions $f_1,\ldots,f_n$ give the gene regulatory rules. If $x_i=1$ the corresponding gene is expressed (**ON**), and if $x_i=0$ the gene is not expressed (**OFF**). The state space graph of a Boolean network $\mathbf{f}$ is a directed graph with the vertices given by the set $\{0,1\}^n$. There is a directed edge from vertex $\mathbf{v}_1$ to vertex $\mathbf{v}_2$ if the value of $\mathbf{f}$ at $\mathbf{v}_1$ is $\mathbf{v}_2$.

3 APPROACH

To explain our approach, we begin by examining the transition from the experimental data to a wiring diagram, and we want to transform the experimental data to the information on the updating functions on certain subsets of the state space $\{0,1\}^n$ instead. We observe that giving a wiring diagram is the same thing as giving the information on the values of each updating function f_i on certain subsets U_i and V_i of $\{0,1\}^n$, for all $1\le i\le n$.

Let us consider the wiring diagram in Figure 1 as an example. The indication that wg_3 is self activated in the wiring diagram in Figure 1 can also be represented by the fact that the expression of wg_3 is 'on' for all those states (sequences of 0 and 1 of length 6), which have a 1 at the third spot. In terms of updating functions, that means the updating function f_3 takes the value 1 for all states of the form $**1***$, where the $*$'s can be either 0 or 1.

We shall assume that f_i takes the value 0 on all the states in U_i, i.e. the expression of corresponding gene is 'off' for these conditions; and takes value 1 on all the states in V_i, i.e. the expression of the corresponding gene is 'on' for these conditions. Since f_i takes different values on U_i and V_i, this pair of sets is disjoint, i.e. $U_i\cap V_i=\emptyset$ (empty set), and this is true for all $1\le i\le n$. If the union $U_i\cup V_i=\{0,1\}^n$, then the values of f_i on all the states are known, and f_i is completely determined. If this happens for all $1\le i\le n$, then all the updating functions are completely determined, and there is only one Boolean network fits the given data. However, if only partial information is known, then $U_i\cup V_i\subsetneq\{0,1\}^n$ for at least one index i, and there are many Boolean networks fit the data. In this case, the number of the Boolean networks that fit the data depends on how much information we know about the biological system; and to choose a Boolean network (or to construct a wiring diagram), we need to make assumptions on those values of the f_i's that have yet to be supported by experimental evidence in order to enlarge the subsets U_i and V_i to, say $\hat{U}_i$ and $\hat{V}_i$ respectively, make the equality $\hat{U}_i\cup\hat{V}_i=\{0,1\}^n$ hold for all $1\le i\le n$.

The other piece of experimental evidence, namely the observed stable states of the biological system, is a subset of states, say S, of the entire state space. To say that a Boolean network given in the form of a polynomial function as in (1) is capable of producing the states of S as its stable states is to say that the states in S are solutions (may or may not be all the solutions) of the system of equations

$$f_i(x_1,x_2,\ldots,x_n)=x_i,\quad 1\le i\le n. \tag{2}$$

That is, they are the so-called *fixed points* of the Boolean network.

This is where the complexity occurs. Though it is possible to check the states in S to see if they are solutions of the system of equations given by (2), in the cases where some or all of the states in S are not solutions of the Boolean network at hand, without further theoretical guidance, it will be difficult to make modifications on the assumptions we made in the construction of the wiring diagram so the new Boolean network will be more consistent with the experimental data, especially for large size networks.

To link the topology of a wiring diagram with the stable states of the corresponding system, we propose to consider n pairs of disjoint subsets U_i, V_i, $1\le i\le n$, of $\{0,1\}^n$, which carry the original information we use to construct a wiring diagram. We also have a set S of observed stable states, which we want our Boolean network to be capable of producing as its stable states. Then our goal is to derive some equations that relate the information carried by the given sets U_i, V_i, $1\le i\le n$, and S.

4 MAIN RESULTS

4.1 Theoretical Results

To describe our first result, we need to construct n pairs of disjoint sets from S as follows. For each index i, let $S_{i,0}$ be the subset of states in S with a 0 at the i-th spot, and let $S_{i,1}$ be the subset of states in S with a 1 at the i-th spot. Note that some of these sets may be empty. For instance, in the case of the Boolean model for the expression pattern of the segment polarity genes in *Drosophila melanogaster*, the set of experimentally observed stable states is $S=\{000101,000011,001100\}$ (Albert and Othmer, 2003), thus for the index $i=1$, since the first spots of the states in S are all 0, we have $S_{1,0}=S$ and $S_{1,1}=\emptyset$.

To distinguish between the set of the stable states that a Boolean network $\mathbf{f}$ produces from the given set S in our statements, we write $S(\mathbf{f})$ for the set of stable states of $\mathbf{f}$. We are now ready to state our first result.

THEOREM 1. *Suppose we are given pairs of disjoint subsets U_i and V_i of $\{0,1\}^n$, $1\le i\le n$. Assume that there is at least one index i, such that the union $U_i\cup V_i\subsetneq\{0,1\}^n$. Then the existence of a Boolean network $\mathbf{f}=(f_1,\ldots,f_n)$ such that the following conditions hold:*

(1) *f_i takes the value 0 on U_i and takes the value 1 on V_i, $1\le i\le n$; and*

(2) *$\mathbf{f}$ is capable of producing the states in S as its stable states, namely $S\subseteq S(\mathbf{f})$,*

is equivalent to the conditions:

$$U_i\cap S_{i,1}=\emptyset \text{ and } V_i\cap S_{i,0}=\emptyset \text{ for all } 1\le i\le n.$$

To give the detail for the proof of this theorem and to describe a method of constructing these Boolean networks if the conditions of the theorem are satisfied, we will need more preliminary information and notation, thus we prefer to provide them in the Supplementary Material.

Next, we describe a condition for the existence of Boolean networks that satisfy the known information on a wiring diagram with precisely S as its stable state set. This condition offers other interesting information. For instance, it gives insight into the robustness of a biological system from a Boolean model: the closer of the Boolean model is to being satisfying the condition the more stable the system is, since in that case, the chance of the system ends up in a stable state other than the ones in S is smaller under perturbations of states (changes of the initial condition or state); while further away from the condition the more sensitive the system is to perturbations, and thus has a greater chance of evolving into new functions (Oikonomou and Cluzel, 2006; Pomerance *et al.*, 2009; Xiao and Dougherty, 2007).

Some extra notation are needed before we can state this result. Let $U_{i,0}$ be the subset of states in U_i with a 0 at the i-th spot, and let $V_{i,1}$ be the subset of states in V_i with a 1 in the i-th spot, $1 \le i \le n$.

THEOREM 2. *Under the assumption that the conditions in Theorem 1 are satisfied, the existence of a Boolean network* $\mathbf{f}$ *such that* $S(\mathbf{f}) = S$ *is equivalent to*

$$\bigcap_{i=1}^{n} (U_{i,0} \cup V_{i,1}) \subseteq S.$$

The proof of this theorem will also be given in the Supplementary Material.

4.2 Algorithm

The theoretical results stated in the two theorems above can be readily implemented, and we now summarize the general procedure provided by the theorems for checking if we can construct a Boolean network from partial information (such as experimental data), which is both capable of producing the given set S of states as its stable states and consistent with the partial information on the interactions among the nodes, or we need to modify some of our assumptions on the network and/or gather further information (such as finding the missing links of interactions).

(1) For each $1 \le i \le n$, define subsets U_i, $U_{i,0}$, V_i, $V_{i,1}$, $S_{i,0}$, and $S_{i,1}$ from the given information as follows.
- The set U_i is all the states such that the i-th updating function f_i takes the value 0 and the set V_i is where f_i takes the value 1.
- Let $U_{i,0}$ be the subset of those states in U_i with a 0 at the i-th spot, and let $V_{i,1}$ be the subset of those states in V_i with a 1 on the i-th spot.
- Let $S_{i,0}$ be the subset of states in S with a 0 at the i-th spot, and let $S_{i,1}$ be the subset of states in S with a 1 at the i-th spot.

(2) Compute the intersections $U_i \cap S_{i,1}$ and $V_i \cap S_{i,0}$ for all $1 \le i \le n$.

(3) Compute $\bigcap_{i=1}^{n} (U_{i,0} \cup V_{i,1})$.

(4) If all the intersections computed in Step (2) are empty, then there exist Boolean networks which both fit the given information on interactions among the nodes and are capable of producing the states in S as their stable states. If not, one can modify the sets U_i and V_i by taking into consideration of further information (the computed result in Step (2) can be used a guide for further data collection).

(5) If all the intersections computed in Step (2) are empty and the intersection computed in Step (3) is a subset of S, then there exist Boolean networks which both fit the given information on interactions among the nodes and are capable of producing *precisely* the states in S as their stable states. If not, one can modify the subsets U_i and V_i, as mentioned in Step (4) as needed, to reduce the size of the intersection computed in Step (3) so that the intersection is a subset of S.

4.3 Examples

We use a simple example to explain the procedure outlined above.

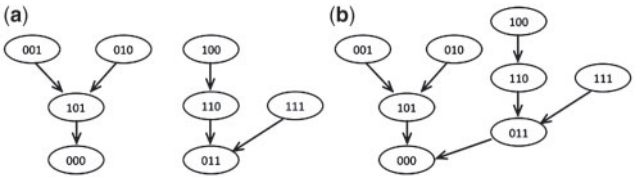

Fig. 2. (**a**) The two diagram on the left give the state space graph for the original model. (**b**) The diagram on the right gives the state space graph for the modified model.

EXAMPLE 4.1. *Assume that the Boolean network we start with is given by the following updating functions:*

$$f_1 = x_1 + x_2 + x_3 + x_1x_2x_3,$$
$$f_2 = x_1 + x_1x_3 + x_2x_3,$$
$$f_3 = x_2 + x_3 + x_1x_3 + x_2x_3 + x_1x_2x_3.$$

This Boolean network has two stable states 000, *and* 011 *(Fig. 2a).*

We remark that in this example, since we start with a given Boolean network, the unions $U_i \cup V_i$ *are always equal to the entire state space (no other choice for a Boolean network unless we change the sets* U_i *and* V_i*).*

(1) *The subsets are*

$$U_1 = \{000, 011, 101, 110, 111\}; U_{10} = \{000, 011\};$$
$$V_1 = \{001, 010, 100\}; V_{11} = \{100\};$$
$$U_2 = \{000, 001, 010, 101\}; U_{20} = \{000, 001, 101\};$$
$$V_2 = \{011, 100, 110, 111\}; V_{21} = \{011, 110, 111\};$$
$$U_3 = \{000, 100, 101\}; U_{30} = \{000, 100\};$$
$$V_3 = \{001, 010, 011, 110, 111\}; V_{31} = \{001, 011, 111\};$$
$$S_{10} = \{000, 011\}; S_{11} = \emptyset; S_{20} = \{000\};$$
$$S_{21} = \{011\}; S_{30} = \{000\}; S_{31} = \{011\}.$$

(2) *It can be seen immediately that all the intersections* $U_i \cap S_{i,1}$ *and* $V_i \cap S_{i,0}$ *are empty as expected.*

(3) *To compute* $\bigcap_{i=1}^{n} (U_{i,0} \cup V_{i,1})$*, we first compute the unions*:

$$U_{10} \cup V_{11} = \{000, 011, 100\};$$
$$U_{20} \cup V_{21} = \{000, 001, 011, 101, 110, 111\};$$
$$U_{30} \cup V_{31} = \{000, 001, 011, 100, 111\}.$$

Then compute the intersection $\bigcap_{i=1}^{n} (U_{i,0} \cup V_{i,1}) = \{000, 011\}$. *The intersection is equal to* S *and hence is a subset of* S *as expected.*

Now suppose we want to change the set of stable states so that the new stable set contains only one state: $S = \{000\}$. *Without any change, the original Boolean network still satisfies the conditions in Theorem 1 and thus is capable of producing* 000 *as one of its stable states. However, it does not satisfy the condition in Theorem 2 since* $\{000, 011\}$ *is no longer a subset of the new set S. There are many possible ways to change the Boolean network so it produces just* 000 *as its stable state. For instance, we can simply move the state* 011 *from* V_2 *to* U_2*, then the new union* $U_{20} \cup V_{21}$ *will no longer contain the state* 011 *and resulting in the intersection*

$\bigcap_{i=1}^{n}(U_{i,0}\cup V_{i,1})=\{000\}$. *If in addition, we also move* 011 *from* V_3 *to* U_3*, then we get a new Boolean network defined by the updating rules:*

$$g_1 = x_1+x_2+x_3+x_1x_2x_3,$$
$$g_2 = x_1(1+x_3+x_2x_3),$$
$$g_3 = x_2+x_3+x_1x_3;$$

with the state space graph as in Fig. 2b.

As another example, one can check that the wiring diagram in Figure 1 satisfies the conditions in Theorem 1 for the set

$$S=\{000101, 000011, 001100\}$$

of the experimentally observed stable states but does not satisfy the condition in Theorem 2. Thus, according to the theorems, the Boolean network constructed according to the diagram is capable of producing these stable states as well as other stable states. This is indeed the case: the corresponding Boolean network produces 10 stable states including the three observed ones. Details of this example are provided in the MAPLE worksheet as part of the Supplementary Material.

In general, since the choice of a Boolean network is not unique due to the fact that we only have partial information on the system, quite often, the Boolean network we construct for the system will produce more stable states than observed experimentally. Whether one wants to modify the network so it produces less stable states depends on other evidence. It could also turn out these extra stable states predict something that has not been observed. For the *Drosophila melanogaster* example here, there is a discussion for these extra stable states in Albert and Othmer (2003).

5 CONCLUDING REMARKS

We have developed a new approach to the problem of constructing a Boolean model for a biological system given partial information on the wiring diagram and a set of stable states. Our approach leads to two theorems that provide criteria for a Boolean network to have both a given network topology and a given set of stable states. It is clear from the statements of our theorems that virtually no computation is needed to verify these conditions, especially for real-word applications, where the given set S (observed stable states) is usually small compare with the size of the whole space $\{0,1\}^n$. For example, the full Boolean network considered in Albert and Othmer (2003) consists 21 parameters, so the state space is of size 2^{21}, while the size of the set of the experimentally observed stable states is 3. Our theorems have applications in many contexts. For instance, if one wants to modify the topology of a wiring diagram, such as adding new links (Bennett and Hasty, 2008; Isalan *et al.*, 2008), one must make sure that the conditions in the theorem are satisfied, otherwise the model will not be able to produce the desired stable states. For another instance, one can use them in detecting the missing links among the genes if the Boolean model constructed according to a wiring diagram does not produce the observed stable states: links must exist so that the conditions in the theorem are not violated.

As the need to consider complex networks becomes inevitable (Purnick and Weiss, 2009; Strogatz, 2001; Tyers and Mann, 2003), developing theoretical foundation for networks has now been brought to the forefront of network study (Porter *et al.*, 2009). Our results here can be viewed as another step in this direction.

Conflict of Interest: none decalred.

REFERENCES

Albert,R. and Othmer,H. (2003) The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.*, **223**, 1–18.

Aldana,M. and Cluzel,P. (2003) A natural class of robust networks, *Proc. Natl Acad. Sci. USA*, **100**, 8710–8714.

Bennett,M.R. and Hasty,J. (2008) Genome rewired. *Nature*, **452**, 824–825.

Bonneau,R. (2008) Learning biological networks: from modules to dynamics. *Nat. Chem. Biol.*, **4**, 658–664.

Brickenstein,M. and Dreyer,A. (2009) PolyBoRi: A Gröbner basis frame-work for Boolean polynomials. *J. Symbolic Comput.*, **44**, 1326–1345.

Choudhary,A. *et al*. (2006) Intervention in a family of Boolean networks. *Bioinformatics*, **22**, 226–232.

Colón-Reyes,O. *et al*. (2004) Boolean monomial dynamical systems. *Ann. Combinatorics*, **8**, 425–439.

Davidson,E.H. and Levine,M.S. (2008) Properties of developmental gene regulatory networks, *Proc. Natl Acad. Sci. USA*, **105**, 20063–20066.

Feist.A.M. *et al*. (2009) Reconstruction of biochemical networks in microorganisms. *Nat. Rev. Microbiol.*, **7**, 129–143.

Hasty,J. *et al*. (2002) Engineered gene circuits. *Nature*, **420**, 224–230.

Isalan,M. *et al*. (2008) Evolvability and hierarchy in rewired bacterial gene networks. *Nature*, **452**, 840–845.

Jarrah,A.S. *et al*. (2007a) Reverse-engineering of polynomial dynamical systems. *Adv. Appl. Math.*, **39**, 477–489.

Jarrah,A.S. *et al*. (2007b) Nested canalyzing, unate cascade, and polynomial functions, *Phys. D*, **233**, 167–174.

Karlebach,G. and Shamir,R. (2008) Modelling and analysis of gene regulatory networks. *Nat. Rev. Mol. Cell Biol.*, **9**, 770–780.

Kauffman,S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.

Kauffman,S. *et al*. (2004) Genetic networks with canalyzing Boolean rules are always stable. *Proc. Natl Acad. Sci. USA*, **101**, 17102–17107.

Kinoshita,S-i. *et al*. (2009) Intrinsic properties of Boolean dynamics in complex networks. *J. Theor. Biol.*, **256**, 351–369.

Laubenbacher,R. and Stigler,B. (2004) A computational algebra approach to the reverse engineering of gene regulatory networks. *J. Theor. Biol.*, **229**, 523–537.

Nochomovitz,Y.D. and Li,H. (2006) Highly designable phenotypes and mutational buffers emerge from a systematic mapping between network topology and dynamic output. *Proc. Natl Acad. Sci. USA*, **103**, 4180–4185.

Oikonomou,P. and Cluzel,P. (2006) Effects of topology on network evolution. *Nat. Phys.*, **2**, 532–536.

Pal,R. *et al*. (2005) Generating Boolean networks with a prescribed attractor structure. *Bioinformatics*, **21**, 4021–4025.

Pomerance,A. *et al*. (2009) The effect of network topology on the stability of discrete state models of genetic control. *Proc. Natl Acad. Sci. USA*, **106**, 8209–8214.

Porter,M.A. *et al*. (2009) Communities in networks. *Not. Am. Math. Soc.*, **56**, 1082–1097.

Purnick,P.E.M. and Weiss,R. (2009) The second wave of synthetic biology: from modules to systems. *Nat. Rev. Mol. Cell Biol.* **10**, 410–422.

Strogatz,S.H. (2001) Exploring complex networks. *Nature*, **410**, 268–276.

Tamura,T. and Akutsu,T. (2009) Detecting a singleton attractor in a Boolean network utilizing SAT algorithms. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, **E92-A**, 493–501.

Tomlin,C.J. and Axelrod,J.D. (2007) Biology by numbers: mathematical modelling in developmental biology. *Nat. Rev. Genet.*, **8**, 331–340.

Tyers,M. and Mann,M. (2003) From genomics to proteomics. *Nature*, **422**, 193–197.

Xiao,Y. and Dougherty,E.R. (2007) The impact of function perturbations in Boolean networks. *Bioinformatics*, **23**, 1265–1273.

Zhang,S-Q. *et al.* (2007) Algorithms for finding small attractors in Boolean networks. *EURASIP J. Bioinform. Syst. Biol.*, **2007**, doi:10.1155/2007/20180.