A general method to find all attractors of multi-level discrete networks

Xiao Gan
Department of Physics
Penn State University
U.S.A
xxg114@psu.edu

Réka Albert
Department of Physics
Penn State University
U.S.A
rza1@psu.edu

Abstract

Analyzing the long-term behaviors (attractors) of dynamic models of biological systems can provide valuable insight. We propose a general method that can find all attractors of discrete dynamical systems by extending a method that finds all attractors of a Boolean network model. The method is based on finding stable motifs, subgraphs whose nodes' states can stabilize on their own. We extended this method from binary states to any finite discrete levels by establishing a multi-level formalism, where a virtual node is created for each level of a multi-level node, and describing each virtual node with a quasi-Boolean function. We then create an expanded representation of the multi-level network, find multi-level stable motifs, and identify attractors in a similar way as in the Boolean case.

We test and validate the algorithm on representative synthetic multi-level networks and on a published biological network. Multi-level stable motifs offer a way to find all attractors without constraints on the update scheme and suggest ways to control which attractor the multi-level network model evolves into.

Key words

Network, discrete dynamic model, attractor, stable motif, network reduction, control

1 Introduction

Dynamic modeling can offer valuable insight in understanding the emergent properties of interacting biological systems (Arenas, Díaz-Guilera, Kurths, Moreno, & Zhou, 2008). For example, it can be used to link the interactions among cellular constituents (e.g. mRNAs, proteins or small molecules) to cell-level functions or behaviors (Barabasi & Oltvai, 2004). In such models, each node is associated with a variable, representing its abundance or concentration. In discrete dynamic models, the state variable of a node is determined from a discrete (logical) regulatory function consisting of regulators of the node. Compared with continuous modeling, discrete modeling has the advantage that it can capture the system's behavior without the need of much kinetic detail (Albert & Wang, 2009). Update scheme, or timing of a discrete model, is also discrete. There are deterministic update schemes like synchronous update where all nodes are updated at each time step according their regulatory function; there are also stochastic update schemes, for example a general asynchronous update where in each time step, one node from the network is randomly chosen to update. The long-term behavior(s) of a dynamic system are called attractors, which include steady-state attractors (fixed points), and complex (oscillating) attractors where a subset of the nodes do not take fixed values. In discrete models, the state space of the system can be represented in a state-transition-graph (STG) where nodes represent states of the system, and edges represent state transitions. The terminal strongly-connected-components (SCCs) of the STG are attractors of the system.

Attractors represent system-level outcomes. For example, in a cell signaling network, attractors can correspond to cell types, cell fates or behaviors. Therefore, finding the attractors of a network model is significant. Most methods that analyze attractors work only for Boolean dynamics. Although for some systems Boolean modeling is appropriate, it is often necessary to have multiple levels for a node. Existing methods that find attractors of multi-level models are rare and may have special constrains when finding complex attractors, for example requiring a synchronous update scheme (Dubrova, Liu, & Teslenko, 2012) and/or requiring a very limited complex attractor length (Hinkelmann et al., 2011). Therefore a general method that can find all attractors of a network model is needed.

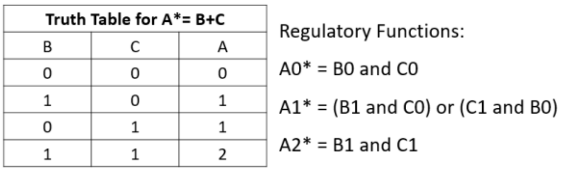
Zanudo &Albert proposed an effective attractor finding method for Boolean models (Zanudo & Albert, 2013). It finds not only steady states, but also possible complex attractors without dependence on the timing of events (or on the update scheme). In this paper, we extend this method from Boolean to any discrete level. We test and validate the method on synthetic networks and on a biological model in the literature. Although this method is developed with a primary motivation to analyze biological networks, it is general and can be applied to any finite discrete dynamical system.

2 Methods

The idea of the method of Zanudo & Albert is: first create an expanded representation of the Boolean network that incorporates all the regulatory functions by introducing composite nodes that express the ‘And’ Boolean operation, and by expressing each state of a node as a separate ‘virtual node’. Stable motifs, subgraphs whose nodes’ states can stabilize on their own, can then be identified from the topology of the expanded network, leading to reduction of the network and identification of attractors. In the following we describe how to adapt this method to the multi-level framework.

2.1 Quasi-Boolean formalism of multi-level models

We establish a formalism where multi-level regulatory functions become Boolean-like. We treat each level (state) of a multi-level node as a separate node, called a virtual node. For example, if a node A has 3 different levels, 0, 1, and 2, then 3 virtual nodes for A, namely A0, A1, A2, are created in our formalism. Each virtual node is like a Boolean variable, and the combination of all virtual nodes represent the state of the original node. We will refer to these virtual nodes as ‘sibling nodes’ of each other. For example, original state A=2 will now be represented as the combination A0=0, A1=0, A2=1. Note that one and only one of the virtual nodes takes value 1, while all other virtual nodes, i.e. its sibling nodes, must all be 0. Then we write the regulatory function of each virtual node in a Boolean disjunctive normal form, by treating each input combination as a conjunctive clause and then connect all conjunctive clauses that yield the same target node level with the Boolean ‘or’ operator. Figure 1 demonstrates the example of the regulatory function A* = B+C. The ‘*’ sign in the function indicate this function determines the state of the target node in the next time step.



Truth Table for A*= B+C		
B	C	A
0	0	0
1	0	1
0	1	1
1	1	2

Regulatory Functions:

A0* = B0 and C0

A1* = (B1 and C0) or (C1 and B0)

A2* = B1 and C1

Figure 1. Demonstration of the construction of a quasi-Boolean regulatory function. A 3-level node A has regulatory function: A* =B+C, where B and C both have 2 levels. From the truth table, one can re-write the regulatory function for each virtual node of A, by connecting all conjunctive clauses that yield the same A level with the Boolean ‘or’ operator. In this way, each virtual node’s regulatory function is obtained in a Boolean disjunctive normal form.

Note that the Boolean ‘not’ rule is absent from this formalism, because we have assigned virtual nodes for all states of nodes. A negation is now replaced with an activation by the sibling states instead. We will proceed through the rest of our analysis based on the regulatory functions of the virtual nodes, instead of the functions of the original nodes.

We require the regulatory functions to be written in a disjunctive normal form with all of their prime implicants present, or in other words, in the Blake canonical form. For example, a regulatory function ‘A* = B and C or D and not C’ should be re-written as ‘A* = B and C or D and not C or B and D’, as the conjunctive clause ‘B and D’ is also a prime implicant of A. In the Boolean framework, the Blake canonical form is obtained by applying the Quine-McCluskey (QM) algorithm. To obtain the Blake canonical form in a multi-level function, we developed a multi-level version of the QM algorithm that resembles its Boolean counterpart[1]. The only difference in the multi-level QM from the Boolean is that, while Boolean QM merges minterms when two (all) states of a specific input variable are present, multi-level QM merges minterms when all states of a specific input variable are present.

2.2 The expanded network representation

A network can be reduced prior to attractor identification without affecting its attractor repertoire. Source nodes are not regulated and will stabilize at a fixed state. They can be reduced from the network by plugging in their values into the regulatory functions of the nodes they regulate. This can be done iteratively until no source nodes are present in the network, without affecting the attractors (Saadatpour, Albert, & Reluga, 2013). For some biological networks, this reduction, combined with reduction of mediator nodes, can reduce a large proportion of the network model, leading to a much simplified model.

After this pre-processing step, we can create an expanded network, which is a representation of the network with regulatory functions embedded. The expanded network is obtained from the original one by the following operations: 1. Include each virtual node in the expanded network, and connect all the virtual node’s regulators to it; 2. for each ‘and’ rule in the regulatory functions, create a composite node, and re-wire the edges from the input nodes of the ‘and’ rule to this composite node, then connect the composite node to the regulated node. The original edges from input nodes of the ‘and’ rule to the target node are removed. Figure 2 exemplifies the construction of an expanded network from a regulatory function. To construct the entire expanded network, all virtual nodes and all interactions must be created.

The expanded network contains not only the network structure, but also all information about the regulatory functions. Furthermore, interactions of a combinatorial nature are separated, as all ‘and’ rules are becoming explicit. In this way, the expanded network makes it easy to identify a sufficient condition to activate a node: a virtual node will have state 1, if any regulator virtual node is 1, or if any composite node has all its input virtual nodes being 1, regardless of the states of the rest of its regulators.

---

[1] The details of the multi-level Quine-McCluskey algorithm are available by request.

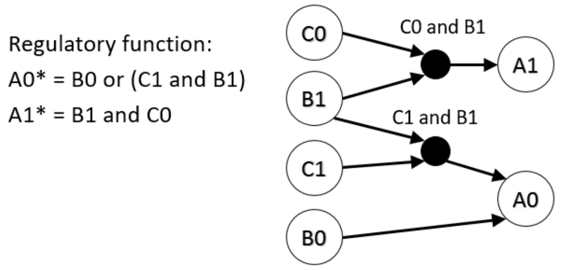


Figure 2. Construction of an expanded network from a regulatory function. Virtual node A0 has function A0* = B0 or (C1 and B1), so in the expanded network, B0 is connected directly to A0; C1 and B1 are connected indirectly to A0 via composite node 'C1 and B1'. A1 has function A1* = C0 and B1, so C0 and B1 are connected indirectly to A0 via composite node 'C0 and B1'.

Following this intuition, a cycle in the expanded network satisfying the above criteria will be self-sufficient to stabilize.

2.3 Stable motifs

A stable motif is a motif in the expanded network that can stabilize on its own. We define it in the following way: a stable motif is a strongly-connected-component (SCC) in the expanded network that satisfies: (1) the SCC contains no sibling node pairs; (2) if the SCC contains a composite node, all of its input nodes must also be in the SCC. The first condition is a natural requirement for a stabilized state for the original node; the second condition is about the nature of the Boolean ‘and’ function, as all inputs must be present to activate the ‘and’ function. For the purpose of the algorithm, we will try to identify stable motifs as the smallest SCCs that satisfy the above condition. In this way we do not miss possible stable motifs when we develop the algorithm. The additional ‘smallest’ condition is not needed for a stable motif to stabilize on its own. Figure 3 shows a three node network example, where the expanded network is constructed and the stable motifs are identified.

In order for stable motifs to be correctly recognized, the regulatory functions must contain all prime implicants. If a prime implicant is missing, a sufficient condition for a node to stabilize is missing, which would lead to false identification of stable motifs. This is why we require the Blake canonical form of functions in the formalism mentioned in 2.1. With Blake canonical normal form of regulatory function, one can prove that there is a one-to-one correspondence between the stable motif and a partial steady state[2]. So by finding stable motifs, we find all steady states and partial steady states of the system.

2.4 Oscillating motifs

An oscillating motif is defined as the largest SCC in the expanded network that satisfies: (1) each virtual node in the SCC must have at least one sibling node in the SCC; (2) if the SCC contains a composite node, all its input nodes must also be in the SCC. Different from the stable motifs, an oscillating node must be able to enter at least two states, so the first condition is necessary. The second condition is also necessary due to the combinatorial nature of the composite node. Since these are necessary but not sufficient conditions for an oscillation, the nodes in an identified oscillating motif are not guaranteed to oscillate. Oscillations are dependent on timing and update scheme. That is, whether an oscillation exists or not depends on the timing of individual events. Based on this fact, it is intuitive that our method cannot find exact oscillations, because our method is independent of timing. We expect that, for every oscillation in the original model, there is an oscillating motif candidate that contains the virtual nodes representing all states visited in that oscillation. In our benchmarks presented in section 3.1, where we used a general asynchronous update scheme, this conjecture was never violated. Figure 4 shows an example of a simple oscillation in multi-level network model.

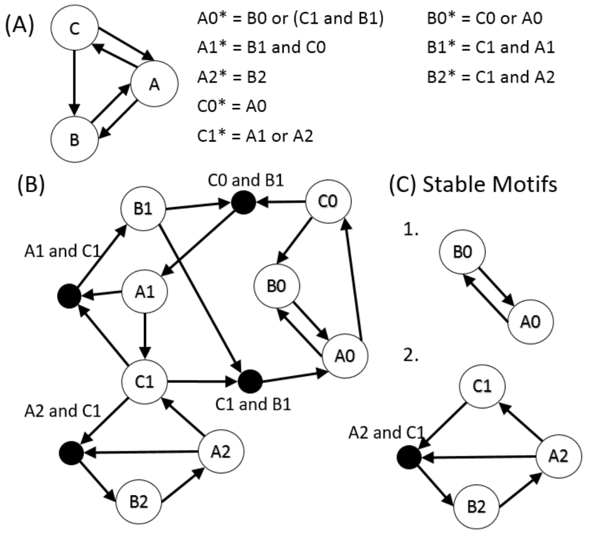


Figure 3. A demonstration of stable motifs identification for a three-node network. (A) The original network and the regulatory functions of each node; (B) the expanded network. It is constructed according to the steps in section 2.2, and then the stable motifs are found by their definition in 2.3. (C) Stable motifs found in this example. The first stable motif, A0, B0, corresponds to a steady state of the system A=0, B=0, C=0. The state C=0 is found by plugging A=B=0 into the regulatory function of C. The 2nd stable motif corresponds to another steady state attractor A=2, B=2, C=0.

2.5 Iterative reduction based on motifs

We plug in the states of the nodes specified in the motifs into the expanded network, and thus reduce the network. For stabilized nodes, the stabilized virtual node takes value 1 and its sibling nodes are set to 0; for oscillating motifs, the nodes included in the oscillating motif are marked as oscillating, and their sibling nodes excluded from the oscillation are set to 0. The downstream nodes of the motif nodes may stablize as a result. In this way, a reduced version of the network model is obtained. We then find stable

---

[2] The details of the proof are available by request.

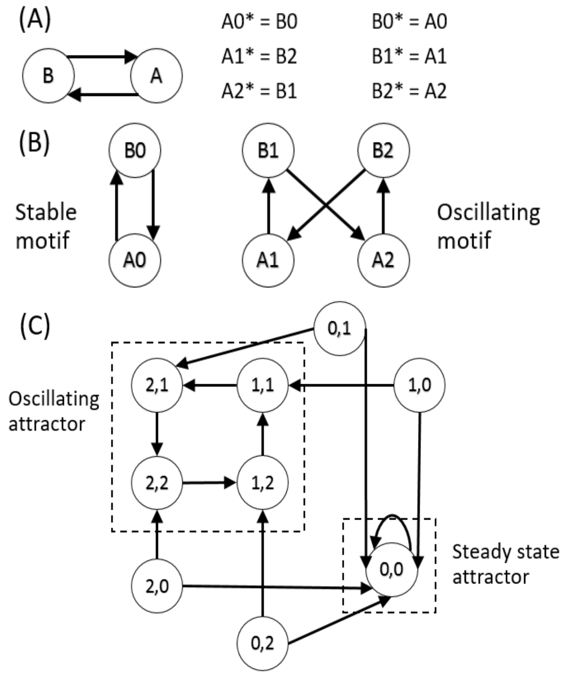


Figure 4. An example of oscillating motif in the multi-level network. A. is the network and regulatory functions; B. the expanded network and motifs. A0 and B0 form a stable motif, indicating a steady state A=0, B=0; while A1, A2, B1 and B2 form an oscillating motif, indicating a possible complex attractor involving states A=1, A=2, B=1 and B=2. The stable motif and oscillating motif identified in 4B correspond to a steady state attractor and an oscillating attractor, respectively. This co-existence of a steady state and an oscillation of the same node does not exist in Boolean models.

motifs and oscillating motifs in the reduced network and plug in the motif values again, until this cannot be done any more. In the end, if all nodes have stabilized, a steady state attractor is found; if there are nodes remaining, those nodes must be in an oscillating attractor. The reduction process can be represented in a motif succesion diagram. Figure 5 demonstrates an example motif succession diagram, where iterative network reduction based on identified motifs leads to identification of attractors.

2.6 Description of the algorithm

Here we give a description of the algorithm[3]. The algorithm requires specific form of input regulatory functions.

1. Reduce the source nodes of the network model by plugging their values into the nodes they regulate. Repeat until no source node is present.
2. Transform the regulatory functions to Blake canonical form using the multi-level Quine-McCluskey algorithm.
3. Create the expanded network according to the definition in 2.2.
4. Search the expanded network for stable motifs and oscillating motifs.
5. For each stable motif and oscillating motif identified, create a copy of the network, with the node states specified in the motif plugged in. Specifically for oscillating motifs, the nodes in the oscillating motif are marked, and their sibling nodes not in the motif are set to 0.
6. Repeat 1, 2, 3, 4, 5 until cannot identify any more motifs. In step 1 the reduction process, marked oscillation virtual nodes are considered as ‘unknown’, and are not reduced when evaluating regulatory functions.
7. Discard duplicate attractors.

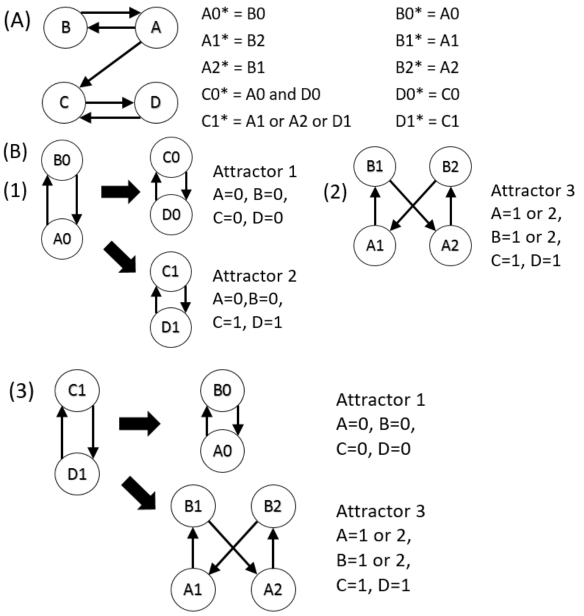


Figure 5: Attractor identification for a four-node network by a motif succession diagram. A. The network and the regulatory function of each node. B. Motif succession diagram. Three motifs are found from the original network, including 2 stable motifs (A0, B0), (C1, D1), and 1 oscillating motif (A1, A2, B1, B2). For each motif, the values of the motif are plugged into the regulatory functions, reducing the network. Then new motifs are identified from the reduced networks. This iteration leads to the ultimate identification of attractors. For example, when the stable motif (A0,B0) is chosen, the network is reduced down to two nodes, C and D, with new regulatory functions C0* = D0, C1*= D1, D0* =C0, D1*=C1. Two new stable motifs, (C0, D0) and (C1, D1) are found in the reduced network, leading to two attractors (A=0, B=0, C=0, D=0) and (A=0, B=0, C=1 D=1). Note that a sequence of stable motifs unique determines an attractor.

3 Results

To test the effectiveness of our method, we apply it to an ensemble of synthetic networks, and a biological network from the literature.

3.1 Benchmark on synthetic network models

We test the method on synthetic networks of different size. To approximate biological models, we first generate Random Boolean Networks where the in-degree is k=2 for each node. Next, we generate the number of states for each node according to an equal distribution of 2 and 3, that is, each node has 50% chance of having 2 states (Boolean), and 50% chance

---

[3] The source code is available by request.

being 3 states. Then we randomly generate a regulatory function among those consistent with the number of regulators and number of states for each node.

To test whether the method finds attractors correctly, we perform simulations similar to Wang et al. (Wang & Albert, 2013). We start from different random initial conditions, and let the system evolve for T_{step} effective time steps. At each time step, one node is randomly chosen and its state is updated according to its regulatory function. If the new state of the node is the same as before, another node will be selected, until the selected node changes state. If no node can reach a new state, a steady state attractor is reached. If no steady state attractor is reached within T_{step} effective time steps, we consider the system being in an oscillating attractor, and then find the attractor from the state transition graph (STG). Note that the sampling method is a heuristic method, and is likely to fail or become inaccurate when the state space is large. For each simulated steady state, we check if it is predicted by the multi-level stable motif algorithm; for each steady state and partial steady state predicted, we check if there is a simulated attractor that contains them. If the predicted and simulated attractors pass both checks, we say they are consistent. As for oscillating attractors, they depend on update schemes, and there is no definite conclusion. The expectation is that an oscillation found by simulation should be a part of a predicted oscillation. If this is indeed the case (in addition to being consistent), we say the results are highly consistent. In all tests, our method shows highly consistent attractors with the sampling method. Table 1 shows time complexity of the algorithm on synthetic networks.

Multi-level Networks				
Size of network	10	15	20	25
Time(s)	0.07	1.1	48	251
Boolean Networks				
Size of network	10	20	30	40
Time(s)	0.07	0.89	74	600

Table 1. Benchmark of our algorithm on synthetic networks of different size (number of nodes). For each size, 50-100 Kauffman networks with connectivity k=2 are generated. For multi-level networks, each node has 50% chance being 2-level and has 50% chance being 3-levels. These parameters are chosen to resemble biological networks. In all benchmark runs, our method shows highly consistent results with the sampling method.

3.2 Application on a biological network in the literature

Next we test our method on a published model of a biological signal transduction network describing stomatal opening in plants (Sun, Jin, Albert, & Assmann, 2014). The model captures different stomatal opening levels in response to different input signals, namely light of different wavelength, carbon dioxide, and the plant hormone abscisic acid. In a previous study (Gan & Albert, 2016), we reduced the original model and analyzed the dynamic repertoire of the model by combining several methods. The reduced model has 32 nodes and 81 edges, with multi-level nodes that can have as many as 6 to 7 states. Even after the reduction, the state space is still very large, and features multi-stability and oscillations, making the model difficult to analyze with conventional tools. We identify the multi-level motifs and find exactly the same attractors with the previous study, including multi-stability and oscillations, for all input signal combinations[4]. The motifs are found in four cycles of the network: 1. the self-regulation cycle of node PMV (plasma membrane voltage) determines three stable motifs, two of which form bi-stability; 2. A two-node cycle representing the regulation of the concentration of cytosolic Calcium, forms one oscillating motif and one stable motif; 3. A two-node cycle regulating the intercellular CO_2 concentration (C_i) forms one stable motif; 4. A three-node NO (nitric oxide) cycle forms one stable motif. Table 2 exemplifies the motifs found under select signal combinations.

Input signal	Motifs	Agreement
Dual beam, medium CO_2, ABA absent	PMV stable motif; Calcium oscillation; (multi-stability absent)	Yes
Dual beam, high CO_2, ABA absent	PMV bistability; Calcium oscillation	Yes
Red light, medium CO_2, ABA present	PMV bistability; Calcium stable motif C_i stable motif NO stable motif (oscillation absent)	Yes

Table 2. Examples of stable motif algorithm result on the stomatal opening model. The first column is the input signal combination of blue light, red light, carbon dioxide (CO_2), and abscisic acid (ABA). The motifs column lists all motifs found, with the conclusion whether multi-stability or oscillation is predicted. The last column states whether the complete attractors is exactly the same as the results from the previous study. Altogether there are 24 input combinations, and in all those cases, our method finds exactly the same attractors as the results from the previous study.

4 Discussion

The complexity of the algorithm is mainly in the identification of cycles. We search for stable motifs from the intersection of simple cycles in the expanded network. Identifying simple cycles in a directed graph is known to be NP-complete, with time complexity $O\big((N+E)(c+1)\big)$ using Johnson's algorithm (Johnson, 1975), where N is the number of nodes, E is the number of edges, and c is the number of directed cycles. The last can grow faster than 2^N for highly connected networks. In addition, the introduction of multi-level nodes dramatically increases the number

---

[4] The complete attractor analysis result is available by request.

of nodes in the expanded network. These facts limit the algorithm's effectiveness on networks with a large size, a high number of levels, or with high connectivity. Typical biological network models have a low connectivity of k=2, and low number of states for each node. In addition, only a proportion of the nodes are in SCCs. So our method can be successfully applied to these networks. For other types of networks, although our method can theoretically work, the computational complexity may be a challenge. We are working on optimizations of the algorithm so it can work on more complex network models. We are also trying to find more necessary conditions of multi-level oscillation to reduce the number of oscillation candidates.

Stable motifs can control the system by driving the system into specific attractors. Zanudo et al. proved that for the Boolean case, a sequence of stable motifs uniquely determines an attractor (Zanudo & Albert, 2015). This means that by driving certain nodes into their states in a stable motif, the network is driven into the corresponding attractor. The same principle applies to multi-level stable motifs as well, and the algorithm to find the nodes that need to be controlled can be adapted as well. This is particularly valuable in biological networks, as the control of stable motif can suggest possible practical methods to switch the system from an undesired attractor to a desired one.

5 Conclusion

In this paper, we propose a method to find all attractors of a network model, by extending an existing method from Boolean to any discrete level. Benchmark has proven the correctness of the method. This method can find all attractors of the network including both steady states and complex attractors, and is theoretically general enough to work on any discrete model. In addition, the stable motifs offer a way to control the attractor of the network model.

Acknowledgements

This project was supported by NSF grants PHY 1205840, IIS 1161007, and PHY 1545832. The authors thank Jorge G. T. Zañudo and Gang Yang for helpful discussions.

References

Albert, R., & Wang, R. S. (2009). DISCRETE DYNAMIC MODELING OF CELLULAR SIGNALING NETWORKS. In M. L. Johnson & L. Brand (Eds.), *Methods in Enzymology: Computer Methods, Part B* (Vol. 467, pp. 281-306). San Diego: Elsevier Academic Press Inc.

Arenas, A., Díaz-Guilera, A., Kurths, J., Moreno, Y., & Zhou, C. (2008). Synchronization in complex networks. *Physics Reports, 469*(3), 93-153. doi: http://dx.doi.org/10.1016/j.physrep.2008.09.002

Barabasi, A.-L., & Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nat Rev Genet, 5*(2), 101-113.

Dubrova, E., Liu, M., & Teslenko, M. (2012). Finding Attractors in Synchronous Multiple-Valued Networks Using SAT-based Bounded Model Checking. *Journal of Multiple-Valued Logic and Soft Computing, 19*(1-3), 109-131.

Gan, X., & Albert, R. (2016). Analysis of a dynamic model of guard cell signaling reveals the stability of signal propagation. *BMC Syst Biol, 10*(1), 78. doi: 10.1186/s12918-016-0327-7

Hinkelmann, F., Brandon, M., Guang, B., McNeill, R., Blekherman, G., Veliz-Cuba, A., & Laubenbacher, R. (2011). ADAM: Analysis of Discrete Models of Biological Systems Using Computer Algebra. *BMC Bioinformatics, 12*(1), 295. doi: 10.1186/1471-2105-12-295

Johnson, D. B. (1975). Finding All the Elementary Circuits of a Directed Graph. *SIAM Journal on Computing, 4*(1), 77-84. doi: 10.1137/0204007

Saadatpour, A., Albert, R., & Reluga, T. C. (2013). A Reduction Method for Boolean Network Models Proven to Conserve Attractors. *Siam Journal on Applied Dynamical Systems, 12*(4), 1997-2011. doi: 10.1137/13090537x

Sun, Z., Jin, X., Albert, R., & Assmann, S. M. (2014). Multi-level modeling of light-induced stomatal opening offers new insights into its regulation by drought. *PLoS Comput Biol, 10*(11), e1003930. doi: 10.1371/journal.pcbi.1003930

Wang, R. S., & Albert, R. (2013). Effects of community structure on the dynamics of random threshold networks. *Physical Review E, 87*(1). doi: 10.1103/PhysRevE.87.012810

Zanudo, J. G., & Albert, R. (2013). An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos, 23*(2), 025111. doi: 10.1063/1.4809777

Zanudo, J. G., & Albert, R. (2015). Cell fate reprogramming by control of intracellular network dynamics. *PLoS Comput Biol, 11*(4), e1004193. doi: 10.1371/journal.pcbi.1004193