Supplementary Information

Global Stabilization of Boolean Networks with Applications to Biomolecular Network Control

Mohammad Reza Rafimanzelat
mr.rafimanzelat@iauda.ac.ir
Department of Electrical Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran.

Contents

The Boolean network model for S. pombe (fission yeast) cell cycle network

The structure of a BN can be represented by a dependency matrix, A, in which $A_{ij} = 1$ if node *j* activates node *i*, $A_{ij} = -1$ if node *j* inhibits node *i*, and $A_{ij} = 0$ if node *j* does not regulate node *i*.

The dependency matrix of the BN model for S. pombe cell cycle network is as follows[1]:

	SK	Cdc2/13	Ste9	Rum1	Slp1	Cdc2/13*	Wee1	Cdc25	PP	Nodes
	-1	0	0	0	0	0	0	0	0	SK
	0	1	-1	-1	-1	0	0	0	0	Cdc2/13
	-1	-1	0	0	0	-1	0	0	1	Ste9
	-1	-1	0	0	0	-1	0	0	1	Rum1
A=	0	0	0	0	-1	1	0	0	0	Slp1
	0	0	-1	-1	-1	0	-1	1	0	Cdc2/13*
	0	-1	0	0	0	0	0	0	1	Wee1
	0	1	0	0	0	0	0	0	-1	Cdc25
	0	0	0	0	1	0	0	0	-1	PP

The following relation shows how the next state is determined from the current state[1]:

$$x_i(t+1) = \begin{cases} 1 & \sum_j A_{ij} x_j(t) > 0 \\ 0 & \sum_j A_{ij} x_j(t) < 0 \\ x_i(t) & \sum_j A_{ij} x_j(t) = 0 \end{cases}$$

where $x_i(t)$ is the state of node *i* at time step *t*. That is, a node will be active in the next time step if its activatory inputs overweigh its inhibitory inputs, and it will be inactive in the next time step if its inhibitory inputs overweigh its activatory inputs, and it will remain in its previous state, otherwise.

The Boolean network model for the GRN underlying mouse myeloid development

The BN model for the GRN underlying mouse myeloid development consists of 11 nodes, each represented by an abbreviated name: GATA-2, GATA-1, FOG-1, EKLF, Fli-1, SCL, C/EBPα, PU.1, cJun, EgrNab and Gfi-1[2]. We assigned these nodes to network variables, x_1 to x_{11}.

The update rules showing how the state of each node is updated in the next time step according to the current values of other nodes, are as follows[2]:

$$\mathrm{GATA\_2} = \mathrm{GATA\_2} \wedge \overline{(\mathrm{GATA\_1} \wedge \mathrm{FOG\_1})} \wedge \overline{\mathrm{PU.1}};$$

$$\mathrm{GATA\_1} = (\mathrm{GATA\_1} \vee \mathrm{GATA\_2} \vee \mathrm{Fli\_1}) \wedge \overline{\mathrm{PU.1}};$$

$$\mathrm{FOG\_1} = \mathrm{GATA\_1};$$

$$\mathrm{EKLF} = \mathrm{GATA\_1} \wedge \overline{\mathrm{Fli\_1}};$$

$$\mathrm{Fli\_1} = \mathrm{GATA\_1} \wedge \overline{\mathrm{EKLF}};$$

$$\mathrm{SCL} = \mathrm{GATA\_1} \wedge \overline{\mathrm{PU.1}};$$

$$\mathrm{C/EBP\alpha} = \mathrm{C/EBP\alpha} \wedge \overline{(\mathrm{GATA\_1} \wedge \mathrm{FOG\_1} \wedge \mathrm{SCL})};$$

$$\mathrm{PU.1} = (\mathrm{C/EBP\alpha} \vee \mathrm{PU.1}) \wedge \overline{(\mathrm{GATA\_1} \vee \mathrm{GATA\_2})};$$

$$\mathrm{cJun} = \mathrm{PU.1} \wedge \overline{\mathrm{Gfi\_1}};$$

$$\mathrm{EgrNab} = (\mathrm{PU.1} \wedge \mathrm{cJun}) \wedge \overline{\mathrm{Gfi\_1}};$$

$$\mathrm{Gfi\_1} = \mathrm{C/EBP\alpha} \wedge \overline{\mathrm{EgrNab}};$$

The Method Used for Generating Random Boolean Networks

We generated a number of random BNs by changing network parameters as follows:

We used 3 different network sizes (n=8, 9, 10), 2 different average in-degrees ($\overline{k_i} = 2, 3$), 2 different proportions of inhibitory links (PIL=1/2, 1/3), 2 different types of update functions and 20 randomizations of link positions which amounted to 480 BNs.

The following two update rules were used as logical functions for state transitions:

Update rule No. 1: In this rule, all activatory inputs to a node are combined using 'OR' whereas the complement of the inhibitory inputs are combined using 'AND' [3]. Thus, the value of a node will be ON at the next time step if at least one of its activatory inputs is currently ON and none of its inhibitory inputs are currently ON. Table S1 demonstrates the update rule No. 1 for a node with two activatory and one inhibitory inputs as in Fig. S1.

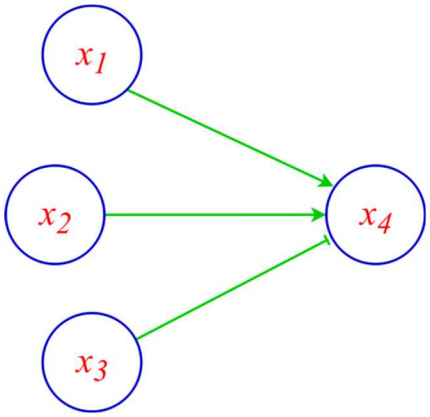

Fig. S1. A BN node with two activatory and one inhibitory inputs

Table S1. Update rule No. 1 represented as a truth table

$x_1(t)$	$x_2(t)$	$x_3(t)$	$x_4(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Update rule No. 2: In this rule, all activatory inputs and the complement of the inhibitory inputs to a node are combined using 'AND' [4]. Thus, the value of a node will be ON at the next time step if all of its activatory inputs are currently ON and none of its inhibitory inputs are ON. Table S2 demonstrates the update rule No. 2 for a node with two activatory and one inhibitory inputs as in Fig. S1.

Table S2. Update rule No. 2 represented as a truth table

$x_1(t)$	$x_2(t)$	$x_3(t)$	$x_4(t+1)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Supplementary References

1. Wang, G. *et al.* Process-based network decomposition reveals backbone motif structure. *Proc. Natl. Acad. Sci.* **107,** 10478–10483 (2010).
2. Krumsiek, J., Marr, C., Schroeder, T. & Theis, F. J. Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PLoS One* **6,** e22649 (2011).
3. Abdi, A., Tahoori, M. B. & Emamian, E. S. Fault diagnosis engineering of digital circuits can identify vulnerable molecules in complex cellular pathways. *Sci. Signal.* **1,** ra10 (2008).
4. Kim, J., Park, S.-M. & Cho, K.-H. Discovery of a kernel for controlling biomolecular regulatory networks. *Sci. Rep.* **3,** 2223 (2013).