Asynchronous Stochastic Boolean Networks as Gene Network Models

Journal:	*Journal of Computational Biology*
Manuscript ID:	Draft
Manuscript Type:	Original Paper
Keyword:	computational molecular biology, GENE NETWORKS, pathways, probability, Statistical models
Abstract:	Logical models have widely been used to gain insights into the biological behavior of gene regulatory networks (GRNs). Most logical models assume a synchronous update of the genes' states in a GRN. However, this may not be appropriate, because each gene may require a different period of time for changing its state. In this paper, asynchronous stochastic Boolean networks (ASBNs) are proposed for investigating various asynchronous state updating strategies in a GRN. As in stochastic computation, ASBNs use randomly permutated stochastic sequences to encode probability. A GRN is considered to be subject to noise and external perturbation, which are investigated by several stochasticity models. Hence, both stochasticity and asynchronicity are considered in the state evolution of a GRN. As a case study, ASBNs are utilized to investigate the dynamic behavior of a T helper network. It is shown that ASBNs are efficient in evaluating the steady state distributions (SSDs) of the network with random gene perturbation. The SSDs found by using ASBNs show the robustness of the attractors of the T helper network, when various stochasticity and asynchronicity models are considered to investigate its dynamic behavior.

Asynchronous Stochastic Boolean Networks as Gene Network Models

Peican Zhu and Jie Han

Abstract—**Logical models have widely been used to gain insights into the biological behavior of gene regulatory networks (GRNs). Most logical models assume a synchronous update of the genes' states in a GRN. However, this may not be appropriate, because each gene may require a different period of time for changing its state. In this paper, asynchronous stochastic Boolean networks (ASBNs) are proposed for investigating various asynchronous state updating strategies in a GRN. As in stochastic computation, ASBNs use randomly permutated stochastic sequences to encode probability. A GRN is considered to be subject to noise and external perturbation, which are investigated by several stochasticity models. Hence, both stochasticity and asynchronicity are considered in the state evolution of a GRN. As a case study, ASBNs are utilized to investigate the dynamic behavior of a T helper network. It is shown that ASBNs are efficient in evaluating the steady state distributions (SSDs) of the network with random gene perturbation. The SSDs found by using ASBNs show the robustness of the attractors of the T helper network, when various stochasticity and asynchronicity models are considered to investigate its dynamic behavior.**

Index Terms—**Stochastic Boolean networks (SBNs), stochasticity, asynchronous state update, gene regulatory networks (GRNs).**

I. INTRODUCTION

IN gene regulatory networks (GRNs), interactions such as the activation and inhibition relationships between genes, protein transcriptional factors and mRNA, are often analyzed using computational models to obtain biological insights. Various methods have been proposed for modeling the genetic interactions; these include Boolean networks (BNs) (Kauffman, 1969), ordinary differential equation (ODE) based methods (Chen et al. 2004; Qian et al., 2008; Li et al., 2008) and stochastic simulations at the molecular level (Gillespie, 1976; Gillespie, 1977). While the stochastic and ODE-based approaches provide a more accurate simulation of a biological system, they also require more detailed knowledge about many parameters *a priori*, such as the kinetic rate constants in a modeling (de Jong, 2002; Karlebach and Shamir, 2008). BNs model the relationships among genes through Boolean logic operations. To incorporate the effect of noise in the interactions of genes and molecules, probabilistic Boolean networks (PBNs) have been proposed and studied (Shmulevich et al., 2002a; Shmulevich et al., 2002b; Shmulevich and Dougherty, 2009). Recently, stochastic Boolean networks (SBNs) have been proposed for an efficient modeling of GRNs (Liang and Han, 2012). As in stochastic computation, an SBN employs random binary bit streams for the analysis of a PBN. To further exploit the simplicity of BNs, stochastic multiple-valued networks (SMNs) have been proposed for an efficient evaluation of gene networks with multiple discrete values (Zhu and Han, 2013).

For simplicity, the Boolean models usually consider a synchronous update of all genes' states in a network (Shmulevich and Dougherty, 2009; Naldi et al., 2007; Remy et al., 2006). In biology, this is not necessarily the case because the expression of a gene is seldom an instantaneous process. Instead it may require a few milliseconds or even up to a few seconds (Greil et al., 2007); this makes a synchronous model less realistic. In fact, an analysis of the stability of attractors has shown that some attractors may disappear in a BN model with gene perturbation (Klemm and Bornholdt, 2005). This indicates that these attractors may be the result of artifacts due to the synchronous updating rules. An asynchronous state update has therefore been suggested to reduce the artifacts. However, a larger number of transitory states are usually required for deriving an attractor with an asynchronous update strategy, thus the determination of attractors becomes particularly difficult for a large network. In this paper, stochastic computational models are proposed for deriving the steady state distributions (SSDs) as estimates of the attractors of an asynchronous regulatory network.

Biological systems are susceptible to stochastic fluctuations, which present an important characteristic of gene networks, because it contributes to determine the SSDs of a cell population (Zhang et al., 2007). In (Ribeiro and Kauffman, 2007; Willadsena and Wiles, 2007), stochasticity is modeled as the perturbation of a gene's state that occurs with certain probability; however, this stochasticity in node (SIN) model over-expresses the effect of noise. A stochasticity in function (SIF) model is therefore proposed in (Garg et al., 2009) and the differences between the SIN and SIF models are explicitly discussed. In a SIN model, the correlation between a node's current expression level and the probability of changing the expression is not taken into consideration, thus a model of stochasticity with propensity parameters (SPP) is proposed in (Murrugarra et al., 2012). In this model, two probabilities are assigned such that each indicates the likelihood that an update leads to an increase or decrease of the current state.

In this paper, asynchronous stochastic Boolean networks (ASBNs) are proposed as asynchronous gene network models. In an ASBN, a number of genes are updated simultaneously or in a different order, as considered in (Harvey and Bossomaier, 1997; Garg et al., 2008; Luo and Wang, 2013; Wu et al., 2009). An asynchronous update module (AUM) is first proposed to implement the asynchronous update of a gene's state. Various asynchronous update strategies that involve a different number of genes to be updated in certain order are then implemented by using randomly generated stochastic sequences. The long run behaviors of synchronous and asynchronous networks are investigated using ASBNs; comparisons between different asynchronous update strategies are performed. The efficiency of the ASBN is evaluated by a comparison to a simulation-based method (Garg et al., 2008). To analyze the long run behavior of an asynchronous network, a so-called time-frame expansion technique (Liang and Han, 2012; Zhu and Han, 2013) is applied to efficiently compute the SSD. The simulation results of a T helper network of 23 genes show the robustness of its attractors under various state-update strategies.

II. Synchronous and Asynchronous Boolean Networks

A Boolean network (BN) can model the relationships among gene nodes by Boolean logic operations (Kauffman, 1969). A probabilistic Boolean network (PBN) of n genes is defined by $G(V, F)$, where $V = \{x_1, x_2, \ldots, x_n\}$ with $x_i \in \{0, 1\}$ for $i \in \{1, 2, \cdots, n\}$, a set of binary nodes, and $F = \{F_1, F_2, \ldots, F_n\}$, a list of sets of update functions: F_i, $i \in \{1, 2, \cdots, n\}$ (Shmulevich and Dougherty, 2009; Kim et al., 2002). For a BN, a node x_i represents the state of gene i; if x_i□=□1 (or 0), then gene i is (or not) expressed. The predictor function F_i contains the rules that determine the next state of gene i. If the quantization level of the gene node is not limited to binary values, a multiple-valued network can be constructed to model the relationship among genes for reducing the inaccuracy incurred by the Boolean simplification (Kim et al., 2002; Li and Cheng, 2010).

A network model is presented in Figure 1 for a deterministic synchronous update strategy. If F_i, $i \in \{1, \cdots, n\}$, consists of several update functions with different selection probabilities, this network becomes a (synchronous) probabilistic network (Shmulevich and Dougherty, 2009; Kim et al., 2002).

Figure 1 A general synchronous gene network model. F_i denotes the update function for gene i, $x_i(t)$ and $x_i(t+1)$ are the states of gene i at time t and $t+1$ respectively, $i \in \{1,\ 2,\ \cdots, n\}$. n is the number of genes in the investigated network.

When asynchronicity is considered, a deterministic-asynchronous probabilistic Boolean network (DA-PBN) assumes that the state of each gene is independently updated according to its own updating period (Shmulevich and Dougherty, 2009). Whether the state of gene i is updated or not at a time is indicated by a binary variable c (either 1 or 0 respectively). The next state of gene i, $x_i(t+1)$, is then given by:

$$x_i(t+1) = \begin{cases} f^{(i)}\big(x_1(t), \cdots, x_n(t)\big) & if\ c = 1 \\ x_i(t) & otherwise \end{cases}, \tag{1}$$

where $f^{(i)}$ is the predictor function in the DA-PBN for gene i; c is a binary variable that determines whether a gene is to be updated or not.

Furthermore, a different number of genes can be selected in an asynchronous update process. The genes can be updated at the same time in one time step or following certain updating order. In (Harvey and Bossomaier, 1997; Garg et al., 2008), only one gene is randomly selected at a time for updating its state, while in (Luo and Wang, 2013), m gene nodes are randomly chosen for a simultaneous state update in one time step, where *m* is a randomly generated number. Even in one time step, however, the update of *m* genes is likely to be asynchronous and the state of a gene depends on the updating order of the inputs of its predictor function. In (Wu et al., 2009), all the gene nodes are asynchronously updated one by one, so an updated state will subsequently affect the other components' states, even within the same time step. Hence, an asynchronous update model is more realistic in modeling biological behaviors. Since the next state of a network is determined by the updated gene and the remaining genes, an updated gene state is determined by the current state vector and the corresponding predictor function (Garg et al., 2008).

III. Stochastic Logic

Stochastic computation has been utilized to perform probabilistic analysis through the application of Boolean logic by encoding a real number or probability into a random binary bit stream (Gaines, 1969). These random binary bit streams introduce inevitable stochastic fluctuations into the computation, thus making the computational result non-deterministic. However, the use of non-Bernoulli sequences of fixed numbers of 1s and 0s for encoding initial input probabilities can reduce the stochastic fluctuation and produce more accurate results than the use of Bernoulli sequences (Han et al., 2013). Moreover, a stochastic computational approach efficiently handles signal correlations in a logic network.

Stochastic computational models are first utilized in stochastic Boolean networks for modeling GRNs (Liang and Han, 2012). To reduce the inaccuracy incurred in a binary analysis, multiple-valued logic has been considered to generalize the stochastic analysis (Zhu and Han, 2013). In a *k*-valued gene network, randomly permuted sequences of fixed numbers of the *k* values are used. The use of the randomly permuted sequences as initial inputs can reduce the inevitable stochastic fluctuation and produce more accurate results than using randomly generated sequences.

The stochastic logic elements used in this paper are listed in Figure 2, including a buffer, an inverter, AND, OR and a multiplexer (MUX) (Liang and Han, 2012). For synchronous and asynchronous networks, stochastic architectures can be constructed for GRNs with update functions implemented by the stochastic logic gates in Figure 2.

Figure 2 Stochastic logic: (a) a buffer, (b) an inverter, (c) an AND gate, (d) an OR gate, (e) an XOR gate, and (f) a multiplexer. Stochastic logic performs a probabilistic analysis by encoding probabilities into random value streams as proportional numbers of different values.

IV. Stochastic Computational Models for Asynchronous Update Strategies

In a biological system, some cells may update their states immediately while it may take longer for other cells to respond. This phenomenon corresponds to a variable reaction time or rate for each cell. To model this asynchronous process, a 2-to-1 multiplexer (MUX) is used to determine whether a gene's state is updated or not, as determined by the value of the control bit S_i, shown in Figure 3. If the control bit S_i is 1, where $i \in \{1, 2, \cdots, n\}$ and n is the number of genes, then the ith gene's state is determined by the input genes' states and the predictor function F_i (i.e. $F_i\big(x_1(t), \cdots, x_n(t)\big)$). If $S_i = 0$, the gene remains at its current state, indicated by the application of the buffer in Figure 3 (i.e. $x_i(t+1) = x_i(t)$). Hence, the stochastic architecture in Figure 3 accurately implements the function of (1).

Figure 3 An asynchronous update module (AUM) for gene i, referred to as AUM_i, consists of a 2-to-1 multiplexer (MUX) with a control bit S_i. F_i is the predictor function for gene i. $x(t) = \{x_1(t), \cdots, x_i(t), \cdots, x_n(t)\}$ with $x_i(t)$

being the state of gene i at time t, while $x_i(t+1)$ indicates the next state of gene i, $i \in \{1,\ 2,\ \cdots, n\}$. n is the number of genes in the investigated network.

Stochastic architectures for asynchronous networks can be constructed with the control sequences of the MUX that are generated according to the updating rules. The state transition matrices (STMs) are usually applied to derive the steady state distribution (SSD) by an iterative evaluation, especially for synchronous models. However, a matrix-based analysis becomes cumbersome to perform due to the size of STMs for a large network; an analysis using STMs becomes even more challenging for asynchronous networks. Hence, the SSD is evaluated by using the so-called time-frame expansion technique (Liang and Han, 2012). By this technique, the temporal evolution of an asynchronous network is simulated using a spatially iterative structure of the asynchronous network model.

Different strategies of updating one gene or m genes synchronously or in a random order are presented as follows:

- The ROG (randomly one gene) model (Garg et al., 2008): At each time step, only one gene node is randomly selected for a state update, while the remaining nodes stay at their present states.
- The RMG (randomly m genes) model (Luo and Wang, 2013): At each time step, m gene nodes are randomly selected and synchronously updated, while the remaining nodes preserve their present values. Here, m can be either a random number or a fixed value. If $m = 1$, the RMG model is simplified to the ROG model.
- The ARO (all genes updated in a random order) model (Wu et al., 2009): At each time step,

all gene nodes are updated; however, the state update of each gene occurs one by one in a random order. The update is carried out immediately and a state change will affect the change of other genes' states, even within the same time step.

- The MRO (m genes updated in a random order) model: At each time step, only m gene nodes are randomly selected for updating their states and each state update occurs in a random order, while the other genes remain at their present states. If $m = 1$, this is simplified to the ROG model. If $m = n$, the result will be the same as the ARO model.

The general architectures are similar for the ROG and RMG models; the only difference lies in the control sequences $\{S_1, S_2, \cdots, S_n\}$ for the 2-to-1 MUXs, where S_i indicates the control sequence for gene $i \in \{1, 2, \cdots, n\}$. For the control sequences of asynchronous update modules (AUMs) in the ROG model, if the kth bit in the control sequence S_i is 1, i.e., $S_{i,k} = 1$, then $\mathrm{S}_{j,k} = 0$ for $j \neq i$, $i, j \in \{1, 2, \cdots, n\}$, $k \in \{1, 2, \cdots, L\}$ with $\sum_{i=1}^{n} S_{i,k} = 1$ as only one gene is selected for updating its state, where L indicates the length of the control sequences. In contrast, the requirement of $\sum_{i=1}^{n} S_{i,k} = m$ must be met for the RMG model as m gene nodes are randomly selected for updating the states using the control sequences.

The generation of the stochastic control sequences for the AUMs is explained next with an example of a 5-gene network using the ROG and RMG models. Let the number of genes to be updated for the RMG model, i.e. m, be 2. For each column of the generated sequence matrices, only one gene is updated while two gene nodes are randomly selected to be refreshed for $j \in \{1, 2, \cdots, L\}$ for the RMG model. At certain time step, let the generated control sequences be C_{ROG} and C_{RMG} for the ROG and RMG models respectively, as illustrated in Figure 4.

Figure 4 An illustrative example of the generated control sequences at a time step for a network of five genes. (a)

The control sequences for the randomly one gene (ROG) model. (b) The control sequences for the randomly m genes (RMG) model. S_i indicates the control sequence for gene i, $i \in [1, 5]$; j is for the jth bit in each sequence, which can be considered as the jth trial of the L experiments and L indicates the length of the stochastic sequences and $j \in \{1, 2, \cdots, L\}$. $S_{i,j} = 0$ indicates that the control bit is 0 for the asynchronous update module (AUM) in Figure 3, so the state of gene i remains, while $S_{i,j} = 1$ means that the next state is determined by the predictor function F_i. For the RMG model, the number of genes to be updated for the jth trail is assumed to be two, thus $\sum_{i=1}^{n} S_{i,j} = 2$ in (b). n is the number of genes in the investigated network.

For the ARO and MRO models, similarly, the evolution of an asynchronous network is simulated by the time-frame expansion technique. In the ARO model, as all genes are updated in a random order, each time step is divided into n sub-steps and only one gene is chosen to be updated at each sub-step while the other genes maintain their present states. Assume at time t, only one gene is updated at each sub-step (n sub-steps in total); hence, after the evolution of n sub-steps, all the genes are updated, followed by the evolution at time $t + 1$. The ARO model is implemented with only one AUM with a control bit of 1 at each sub-step for the gene to be updated (the AUM is simplified to a buffer as the control bit is 0 for the other genes). If m genes are to be updated in a different order, there will be n-m buffers (or AUMs with a control bit of 0) for the n sub-steps. This becomes an MRO model.

For the ARO model, at each sub-step only one gene is updated while the others preserve their present states. If $S_{i,k} = 1$, $i \in \{1, 2, \cdots, n\}$, $k \in \{1, 2, \cdots, L\}$, the ith gene's state is updated by the predictor function; otherwise, the current state remains. For the MRO model, the total number of 1's for the n sub-steps at time step i is equal to the number of genes to be updated for the MRO model. All genes remain at their present states if the control bits for the AUMs are 0 at a time step. The updating order is indicated by the different position of 1's in the generated sequences.

An example of the random updating order for the AUMs at time i is illustrated for a network of five genes. Following the previous analysis, time step i is divided into five sub-steps. The control sequences of C_{ARO} and C_{MRO} for the ARO and MRO models are shown in Figure 5 (a) and (b) respectively.

Figure 5 An illustrative example of the generated control sequences at a time step for a network of five genes. (a) The control sequences for the all genes updated in a random order (ARO) model. (b) The control sequences for the m genes updated in a random order (MRO) model. t is for the tth time step and $t_1, \cdots, t_5$ are the sub-steps at time t. ${C_{ARO}}_t$ and ${C_{MRO}}_t$ are the generated control sequences at the sub-steps of time t. An entry of 0 indicates the state of gene i remains while a value of 1 indicates that the next state is determined by the predictor functions. The number of genes to be updated at time t is three (in a random order), thus $\sum {C_{MRO}}_t = 3$ in (b).

The updating order of the ARO model at time t is given by the control sequences in Figure 5(a), i.e., $x_1 \rightarrow x_4 \rightarrow x_5 \rightarrow x_3 \rightarrow x_2$, while the updating order of the MRO model at time step i is $x_3 \rightarrow x_5 \rightarrow x_1$, as given by the control sequences in Figure 5(b). For genes x_2 and x_4, the present states at time t remain the same.

V. Asynchronous Stochastic Boolean Networks (ASBNs)

Any gene node in a GRN can change the expression level due to noise or a gene perturbation that occurs with certain probability. The perturbation probability can be encoded as a perturbation flag vector γ with each element indicating whether a gene is to be perturbed or not. When a perturbation occurs, the state of the perturbed gene is determined by the present state and the perturbation flag vector. In the stochasticity in node (SIN) model (Garg et al., 2009), therefore, the effect of perturbation can be modeled by an XOR gate (Liang and Han, 2012), by which the gene state at time t will be flipped with a probability of p, as shown in Figure 6(a).

Figure 6 Stochastic architectures for the models of stochasticity in node (SIN) and stochasticity with propensity parameter (SPP). (a) The stochastic architecture of the SIN model for a binary gene node (Liang and Han, 2012). (b) The stochastic architecture of the SPP model for updating the state of a gene.

In the SIN model, the correlation between a gene node's current expression level and the probability of changing the expression value is not taken into consideration. Even if the expression level of the input node for an update function guarantees the activation or degradation, there still exists a probability that the process will not occur due to stochasticity. Hence, a stochasticity model is proposed in (Murrugarra et al., 2012) with propensity parameters,

$$F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^{n}, \tag{2}$$

where f_i indicates the update function for node x_i, $0 \leq p_i^{\uparrow} \leq 1$ and $0 \leq p_i^{\downarrow} \leq 1$ denote the activation propensity and degradation propensity, and n is the number of genes. A model of stochasticity with propensity parameters (SPP) is described by (Murrugarra et al., 2012)

$$x_i \to f_i(x) = \begin{cases} p_i^{\uparrow} & x_i < f_i(x) \\ p_i^{\downarrow} & x_i > f_i(x) \\ 1 & x_i = f_i(x) \end{cases} \tag{3}$$

In order to implement the function of (3), a multiplexer is used here to simulate the behavior of stochasticity with parameters. A general structure is shown in Figure 6(b).

For the architecture in Figure 6(b), the output is determined by the value in the control sequence S. S consists of ternary values {0, 1, 2}, which indicate the three cases of $S_i(t) < S_{f_i}(t)$, $S_i(t) = S_{f_i}(t)$ and $S_i(t) > S_{f_i}(t)$ for any $i \in \{1, 2, \cdots, n\}$. $S_i(t)$ and $S_i(t+1)$ are the stochastic sequences encoding the probabilities of gene i at time t and $t+1$ respectively. $S_{f_i}(t)$ is the stochastic sequence encoding the expected probability for the state of gene i at time t. This probability is calculated from $f_i(x(t))$, where f_i indicates the update function for gene i and $x(t) = \{x_1(t), \cdots, x_i(t), \cdots, x_n(t)\}$ with $x_i(t)$ being the state of gene i at time t. $S_{p_i^\uparrow}$ and $S_{p_i^\downarrow}$ represent the stochastic sequences encoding the propensity parameters.

Let $(S_i(t))_j$, $(S_{f_i}(t))_j$ and $(S_i(t+1))_j$ be the jth value in $S_i(t)$, $S_{f_i}(t)$ and $S_i(t+1)$. For a ternary case, if $(S_i(t))_j = (S_{f_i}(t))_j$, the output $(S_i(t+1))_j$ is the same as $(S_i(t))_j$, because the transition $(S_i(t))_j \rightarrow (S_{f_i}(t))_j$ occurs with a probability of 1. However, if $(S_i(t))_j < (S_{f_i}(t))_j$, $(S_i(t+1))_j$ is determined by the output of a 2-to-1 multiplexer with $S_i(t)$ and $S_{f_i}(t)$ as input sequences and $S_{p_i^\uparrow}$ as the control sequence. If $(S_i(t))_j = 0 < (S_{f_i}(t))_j = 1$, for example, then $(S_i(t+1))_j = 1$ with a probability of $p_i^\uparrow$. If $\left(S_i(t)\right)_j > (S_{f_i}(t))_j$, then $(S_i(t+1))_j = (S_{f_i}(t))_j$ with a probability of $p_i^\downarrow$ or $(S_i(t+1))_j = \left(S_i(t)\right)_j$ with a probability of $1 - p_i^\downarrow$. These functions are implemented by the 2-to-1 multiplexers in the model in Figure 6(b). Let a and b be the values in $(S_i(t))_j$ and $(S_{f_i}(t))_j$ respectively; the next state of the gene i is obtained as

$$S_i(t+1) = \begin{cases} S_i(t),\ if\ a = b \\ S_{f_i}(t)\ with\ p_i^\uparrow, or\ S_i(t)\ with\ 1 - p_i^\uparrow, if\ a < b \\ S_{f_i}(t)\ with\ p_i^\downarrow, or\ S_i(t)\ with\ 1 - p_i^\downarrow,\ if\ a > b \end{cases}, \quad (4)$$

which is equivalent to the function of (3). This shows that the stochastic model in Figure 6(b) accurately computes the function of the SPP model.

In order to analyze the long run behavior of gene networks with different stochasticity effects, synchronous and asynchronous stochastic Boolean networks (ASBNs) are presented in Figure 7 for the SIN and SPP models. The synchronous network is obtained by implementing the stochasticity models (either the SIN in Figure 6(a) or the SPP in Figure 6(b)) into the architecture in Figure 1. For simplicity, only the randomly one gene (ROG) model is illustrated in Figure 7(b) for an ASBN. For randomly m gene (RMG) or a different updating order, the stochastic networks can be obtained by applying a stochasticity model (either the SIN in Figure 6(a) or the SPP in Figure 6(b)) into the stochastic architecture in Figure 7. Although discussed within the context of BNs, multiple-valued networks such as those in (Zhu and Han, 2013) can also be used to construct an asynchronous network. In a stochastic network, randomly permuted sequence and perturbation vectors are generated for different signal probabilities. With these stochastic sequences as inputs to the stochastic architectures, the stochastic behavior of a gene network can be efficiently analyzed.

Figure 7 Synchronous and asynchronous stochastic Boolean networks (ASBNs) for different stochasticity models. (a) A synchronous SBN. (b) An ASBN for the randomly one gene (ROG) model. F_i: predictor function for gene i, $i \in \{1,\ 2,\ \cdots, n\}$. n is the number of genes in the investigated network. The detailed structures and inputs of the SIN and SPP modules are shown in Figure 6 (a) and (b) respectively.

VI. Results and Discussion

A T helper network of 23 genes with 35 regulatory interaction relationships (Figure 8) (Mendoza and Xenarios, 2006) is considered to show the robustness of the attractors under the various asynchronous update schemes. The functional molecules or molecular complexes are listed as TCR, $NFAT$, $IFN-\beta$, $IFN-\beta R$, $L-18$, $IL-18R$, $IRAK$, $SOCS1$, $IL-12$, $IL-12R$, $STAT4$, $T-bet$, $IFN-\gamma$, $IFN-\gamma R$, $JAK1$, $STAT1$, $IL-4$, $IL-4R$, $STAT6$, $GATA3$, $IL-10$, $IL-10R$, $STAT3$.

Figure 8 A T helper network. A positive regulatory interaction is indicated by a regular arrow, while a negative interaction is represented by a blunted arrow (Mendoza and Xenarios, 2006).

There is no input for four gene nodes, namely, $IL-12$, $IFN-\beta$, $IL-18$ and TCR, in the T helper network. As in (Mendoza and Xenarios, 2006), hence, these four elements are considered to be consistently at state 0. The network in Figure 8 can be implemented by the BN model shown in Figure 9.

Figure 9 A deterministic BN model for the 23-gene T helper network, with $X_1=0$, $X_3=0$, $X_5=0$ and $X_9=0$. X_i, $i \in \{1,2,\cdots,23\}$, indicates a gene node in Figure 8.

The cell population may fall into different subgroups on exposure to input stimuli – this is referred to as a cellular differentiation (Kaern et al., 2005). For the 23-gene T helper network in Figure 8, $IL-2$ functions as a key cytokine in the $Th0$ to $Th1$ differentiation process while a large perturbation of $IL-4$ forces the network to switch from state $Th0$ to $Th2$, as illustrated by Figure 10.

Figure 10 Differentiation of T helper (adapted from (Kaern et al., 2005)). $Th1$ cells express $IFN-\gamma$ and $T-bet$, while $Th2$ expresses $GATA3$ and $IL-4$.

In the state space of the T helper network, the basins draining to different attractors are not evenly distributed; the distribution of initial states leading to the attractors is shown in Table 1. As revealed in Table 1, a random initial state evolves into $Th1$ with the largest probability for each state update strategy. The attractor $Th0$ is reached with the lowest probability. For the asynchronous update strategies, the probability of reaching $Th0$ is even lower than that for the synchronous state update.

The effect of perturbation on different gene nodes has been shown by experiments of the differentiation of T helper cells (Agnello et al., 2003; Bergmann et al., 2001). The stochastic simulations using synchronous and asynchronous update strategies correctly identify the attractors for a wild type T helper network, in which no perturbation exists. Table 2 shows the steady states found by the stochastic approach for several scenarios of the wild type (no gene suppression or overexpression; the initial states are randomly generated with a probability of 0.5), $IL-12$ overexpressed and $IL-4$ overexpressed. These steady states are good estimates of the attractors found in (Garg et al., 2008; Murphy and Reiner, 2002).

In (Garg et al., 2009), at most one perturbation is considered to occur at a time. In this work, k time frames are considered with the stochasticity models (k = 5 in Figure 11). If the initial state of the T helper network is $Th0$, after the constant activation of $IL-12$, the state will transition into $Th1$. As shown in Figure 11 (b) and (c), the state remains even after the inactivation of $IL-12$ if no perturbation occurs. For the SIN model, the $Th0$ cells derive into $Th1$ and $Th2$, while a few cells remain at their current states. The almost equally-likely cellular differentiation of steady states has been analyzed in (Kadanoff et al., 2003). However, it has been shown that a $Th0$ cell cannot transfer into a $Th2$ cell (Murphy and Reiner, 2002). Hence, the SIN model could be the main reason for this discrepancy, because $Th1$ and $Th2$ are evolved from $Th0$ with nearly equal probability, as revealed in Figure 11(d)-(f) for the synchronous and RMG models.

Figure 11 Transitions between attractors during the differentiation process of the T helper network with an external stimulus of $IL-12$ for the synchronous and randomly m genes (RMG) models. Each dot indicates a T helper cell and each cell is independent of the neighboring cells. A red dot represents an undifferentiated cell at $Th0$, a green dot indicates the cell state of $Th1$ and a blue dot indicates the state of $Th2$. (a) All cells are in the initial $Th0$ state. (b) and (c) The transitions from $Th0$ to $Th1$ by synchronous and asynchronous updating rules (without perturbation to the network). (d), (e) and (f) The transitions from $Th0$ for the stochasticity in node (SIN) model with perturbation probability of 0.5 for each node: the percentage of the corresponding nodes indicates the percentage of the cells deriving into an attractor. For the first five time frames, each of the gene nodes is subject to the effect of SIN. The obtained transition probabilities from $Th0$ to different attractors are similar to the values in Table 1, because it is equivalent to setting the initial state as the wile type by having each of the nodes under SIN for several time frames.

For the SPP model, the next state of a gene is closely related with the activity of other nodes, the predictor function and the present states. The simulation results for this model are shown in Table 3 for an initial state of $Th0$. As revealed in the results, nearly all the nodes of $Th0$ differentiate into $Th1$ cells under the dosage of $IL-12$. Similar results can be obtained for the dosage of $IL-4$ and for the other asynchronous update strategies. The instability of $Th0$ is in agreement with the experimental results in (Kaern et al., 2005).

VII. CONCLUSION

In this paper, asynchronous stochastic Boolean networks (ASBNs) are proposed for an efficient modeling of gene regulatory networks (GRNs). Various asynchronous updating rules are considered, including the models of randomly one gene (ROG), randomly *m* genes (RMG), all genes updated in a random order (ARO) and *m* genes updated in a random order (MRO). In an ASBN, if the control bits for an asynchronous update module are all 0's, an asynchronous network becomes the same as a synchronous update network. Stochastic analyses are carried out for the asynchronous models of stochasticity in node (SIN) and stochasticity with propensity parameters (SPP). The expected long run behavior of a GRN is studied by the stochastic approach for various asynchronous state update strategies.

For an asynchronous gene network, an accurate analysis becomes difficult due to the limitation of computational resources, especially when the number of genes or the quantization level of gene states increases. In the stochastic approach, a simulation-based time-frame expansion technique is used for an efficient analysis of the SSD of a network. The obtained SSDs can be used to estimate the attractors of, especially, a large network. This is shown by the study of a T helper network of 23 genes. The robustness of the attractors for the T helper network is demonstrated for a number of synchronous and asynchronous state update strategies.

The effect of different asynchronous updating rules on the distribution of initial states leading to an attractor is investigated for the T helper network. It is shown that the model of stochasticity with propensity parameters (SPP) accurately reveal the state transition from $Th0$ to $Th1$ under the dosage of $IL-12$.

References

[1] Agnello, D., Lankford, CSR., Bream, J., Morinobu, A., Gadina, M., O'Shea, J., Frucht, DM. 2003. Cytokines and transcription factors that regulate T helper cell differentiation: new players and new insights. J Clin Immunol, 23:147-161.

[2] Bergmann, C., Van Hemmen, J. L., Segel, L. A. 2001. Th1 or Th2: how an appropriate T helper response can be made. Bull. Math. Biol. 63, 405–430.10.1006/bulm.2000.0215.

[3] Chen, K. C. et al. 2004. Integrative analysis of cell cycle control in budding yeast. Mol. Biol. Cell, 15, 3841–3862.

[4] de Jong, H. 2002. Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol,* January 2002, 9(1):67-103.

[5] Gillespie, DT. 1976. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J Comput Phys 1976, 22:403.

[6] Gillespie, DT. 1997. Exact stochastic simulation of coupled chemical reactions. J Phys Chem 1977, 81:2340–2361.

[7] Garg, A., Mohanram, K., Di Cara, A., De Micheli, G., and Xenarios, I. 2009. Modeling stochasticity and robustness in gene regulatory networks. Bioinformatics (Oxford, England), 25(12), i101-9.

[8] Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., and De Micheli G. 2008. Synchronous versus asynchronous modeling of gene regulatory networks. Bioinformatics 24, 1917– 1925.

[9] Greil, F., Drossel, B., Sattler, J. 2007. Critical Kauffman networks under deterministic asynchronous update. New Journal of Physics 9:373.

[10] Han, J., Chen, H., Liang, J., Zhu, P., Yang, Z. and Lombardi, F. 2013. A Stochastic Computational Approach for Accurate and Efficient Reliability Evaluation. Accepted for publication in IEEE Transactions on Computers.

[11] Harvey, I. and Bossomaier, T. 1997. Time out of joint: Attractors in asynchronous random Boolean networks, in Proceedings of the Fourth European Conference on Artificial Life (ECAL97, MIT Press), pp. 67–75.

[12] Gaines, B. R. 1969. Stochastic Computing Systems. Advances in Information Systems Science, Vol. 2, pp. 37-172.

[13] Kadanoff, L. et al. 2003. Boolean dynamics with random couplings. Springer Applied Mathematical Sciences Series, Special volume, Springer, New York, pp. 23–89.

[14] Kaern, M. et al. 2005. Stochasticity in gene expression: from theories to phenotypes. Nat. Rev. Genet., 6, 451–464.

[15] Karlebach, G., and Shamir, R. 2008. Modelling and analysis of gene regulatory networks. Nat Rev Mol Cell Biol 2008, 9:770-780.

[16] Kauffman, S. A. 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. Theor. Biol., vol. 22, pp. 437–467.

[17] Kim, S., Li, H., Dougherty, E. R., et al. 2002. Can Markov chain models mimic biological regulation? Journal of Biological Systems, vol. 10, no. 4, pp. 337–357.

[18] Klemm, K., and Bornholdt, S. 2005. Stable and unstable attractors in Boolean networks. Physical Review E, Statistical, Nonlinear, and Soft Matter Physics 72: 055101.

[19] Li, S. et al. 2008. A quantitative study of the division cycle of Caulobacter crescentus stalked cells. PLoS Comput. Biol., 4, e9.

[20] Li, Z., Cheng, D. 2010. Algebraic Approach To Dynamics of Multivalued Networks. International Journal of Bifurcation an d Chaos, Vol. 20, No. 3 (2010) 561– 582.

[21] Liang, J., and Han, J. 2012. Stochastic Boolean networks: an efficient approach to modeling gene regulatory networks. BMC Syst. Biol., 6, 113.

[22] Luo, C., and Wang, X. 2013. Dynamics of Random Boolean Networks under Fully Asynchronous Stochastic Update Based on Linear Representation. PloS one, 8(6), e66491.

[23] Mendoza, L., and Xenarios, I. 2006. A method for the generation of standardized qualitative dynamical systems of regulatory networks. Theor. Biol. Med. Model., 3, 13.

[24] Murrugarra, D., Veliz-Cuba, A., and Aguilar, B., et al. 2012. Modeling stochasticity and variability in gene regulatory networks. EURASIP J. Bioinform. Syst. Biol., 2012 (1), 5.

[25] Murphy, K. M., and Reiner, S. L. 2002. The lineage decisions on helper T cells. Nat. Rev. Immunol., 2, 933–944.

[26] Naldi, A. et al. 2007. Decision diagrams for the representation and analysis of logical models of genetic networks. LNCS/LNBI, 4695, 233–247.

[27] Qian, L., Wang, H., and Dougherty E. 2008. Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering. IEEE Trans. Signal Process., vol.56, no. 7, pp. 3327–3339.

[28] Remy, E. et al. 2006. From logical regulatory graphs to standard petri nets: dynamical roles and functionality of feedback circuits. Springer LNCS, 4230, 56–72.

[29] Ribeiro, A. S., and Kauffman, S. A. 2007. Noisy attractors and ergodic sets in models of gene regulatory networks. J. Theor. Biol., 247, 743–755.

[30] Shmulevich, I., Dougherty, E. R., and Zhang, W. 2002a. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. Proc. IEEE, vol. 90, pp. 1778–1792.

[31] Shmulevich, I., Dougherty, E. R., and Zhang, W. 2002b. Gene perturbation and intervention in probabilistic Boolean networks. Bioinformatics, 18(10):1319-1331.

[32] Shmulevich, I., Dougherty, E. R. 2009. Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks. SIAM, Philadelphia.

[33] Willadsena, K., and Wiles, J. 2007. Robustness and state-space structure of Boolean gene regulatory models. J. Theor. Biol., 249, 749–765.

[34] Wu, M., Yang, X., and Chan, C. 2009. A dynamic analysis of IRS-PKR signaling in liver cells: a discrete modeling approach. PLoS One, 4(12), e8040.

[35] Zhang, S. et al. 2007. Simulation study in probabilistic Boolean network models for genetic regulatory networks. Int. J. Data Min. Bioinformatics, 1, 217–240.

[36] Zhu, P., and Han, J. 2013. Stochastic Multiple-Valued Gene Networks. Accepted for publication in IEEE Transactions on Biomedical Circuits and Systems.

Table 1 The distribution of initial states (in percentages) leading to the attractors of a wild type T helper cell: 200,000 states are randomly chosen for simulation.

Wild type	$Th0$ (%)	$Th1$ (%)	$Th2$ (%)
Synchronous	6.86	58.65	36.89
ROG	3.75	50.07	46.19
RMG (m = 10)	2.94	51.42	45.65
RMG (m = 21)	5.32	54.51	40.17
ARO	3.06	50.70	46.25
MRO (m = 10)	3.79	50.21	46.01

Table 2 Steady states of the T helper network found by the stochastic approach.

Perturbed genes	Active genes in steady states	attractors
Wild type	All genes are inactive	$Th0$
	$IFN-\gamma$ $T-bet$ $SOCS1$ $IFN-\gamma R$	$Th1$
	$IL-10$ $IL-10R$ $GATA3$ $STAT3$ $STAT6$ $IL-4$ $IL-4R$	$Th2$
$IL-12$ over expressed	$IFN-\gamma$ $T-bet$ $SOCS1$ $IFN-\gamma R$ $IL-12$ $IL-12R$ $STAT4$	$Th1$
	$IL-10$ $IL-10R$ $GATA3$ $STAT3$ $IL-12$ $STAT6$ $IL-4$ $IL-4R$	$Th2$
$IL-4$ over expressed	$IFN-\gamma$ $T-bet$ $SOCS1$ $IFN-\gamma R$ $IL-4$	$Th1$
	$IL-10$ $IL-10R$ $GATA3$ $STAT3$ $STAT6$ $IL-4$ $IL-4R$	$Th2$

Table 3 The transition probability from the initial state $Th0$ to an attractor using the synchronous and random m genes (RMG) models with an external stimulus of $IL-12$. If $m=1$, the RMG model is simplified to the random one gene (ROG) model. The propensity parameters $p_i^{\uparrow}$ and $p_i^{\downarrow}$ are assumed to be 0.5 for each node in the model of stochasticity with propensity parameters (SPP).

Update strategies	$Th0 \rightarrow Th1$	$Th0 \rightarrow Th2$
synchronous or RMG (m = 23)	1	0
RMG (m = 10)	1	0
ROG or RMG (m = 1)	1	0

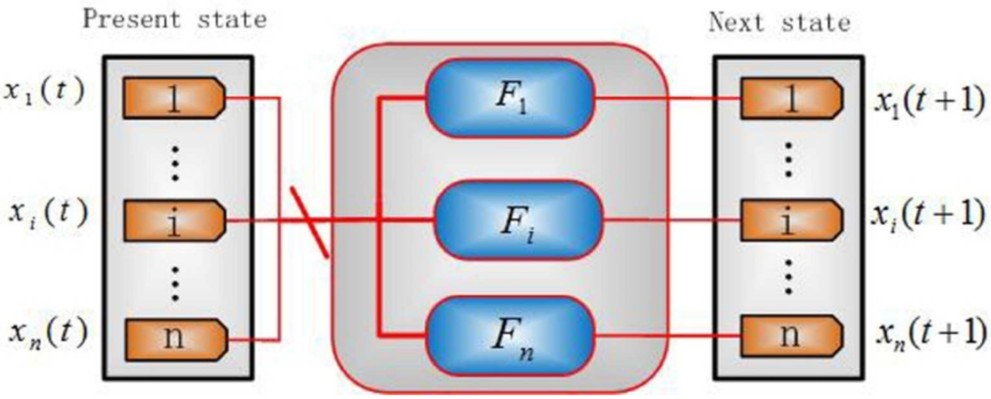

Figure 1 A general synchronous gene network model. F_i denotes the update function for gene i, x_i (t) and x_i (t+1) are the states of gene i at time t and t+1 respectively, i∈{1,2,⋯,n}. n is the number of genes in the investigated network.
59x23mm (300 x 300 DPI)

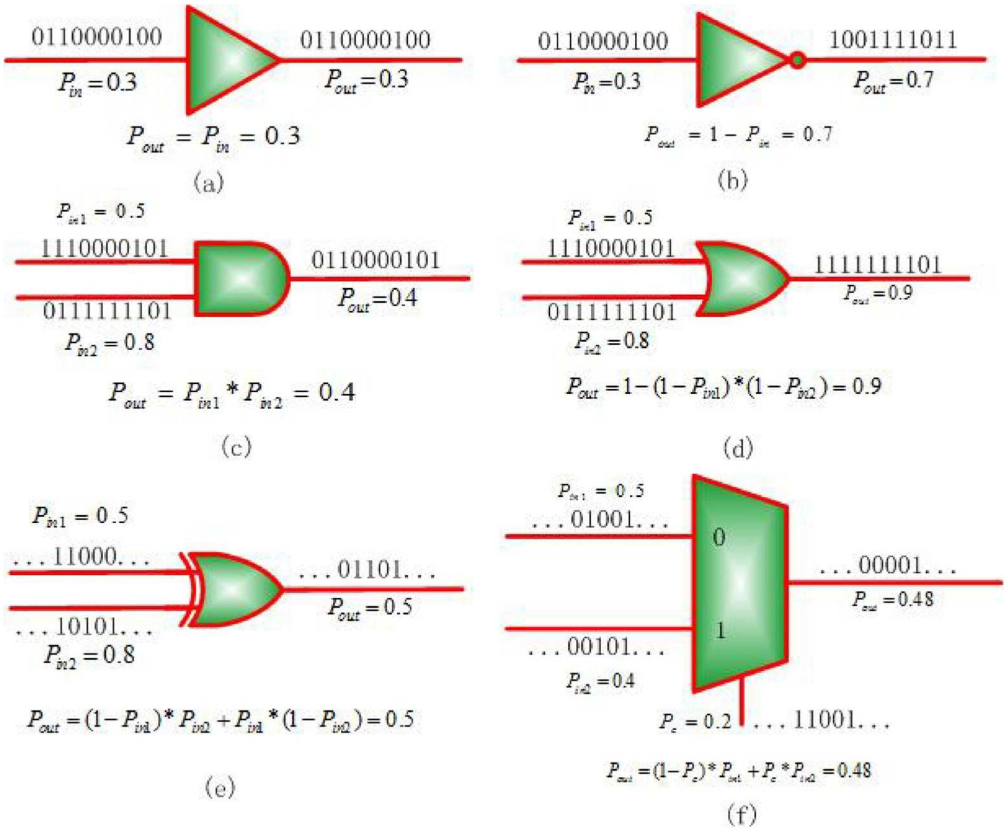

Figure 2 Stochastic logic: (a) a buffer, (b) an inverter, (c) an AND gate, (d) an OR gate, (e) an XOR gate, and (f) a multiplexer. Stochastic logic performs a probabilistic analysis by encoding probabilities into random value streams as proportional numbers of different values.
134x111mm (300 x 300 DPI)

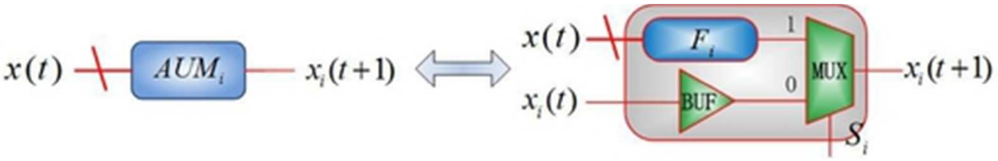

Figure 3 An asynchronous update module (AUM) for gene i, referred to as 〖AUM〗_i, consists of a 2-to-1 multiplexer (MUX) with a control bit S_i. F_i is the predictor function for gene i. x(t)={x_1 (t),⋯,x_i (t),⋯,x_n (t)} with x_i (t) being the state of gene i at time t, while x_i (t+1) indicates the next state of gene i, i∈{1,2,⋯,n}. n is the number of genes in the investigated network.
43x6mm (300 x 300 DPI)

$$C_{ROG}=\begin{bmatrix}S_1\\S_2\\S_3\\S_4\\S_5\end{bmatrix}=\begin{array}{c}j\\ \begin{bmatrix}\cdots&0&\cdots\\ \cdots&0&\cdots\\ \cdots&1&\cdots\\ \cdots&0&\cdots\\ \cdots&0&\cdots\end{bmatrix}\end{array}$$

(a)

$$C_{RMG}=\begin{bmatrix}S_1\\S_2\\S_3\\S_4\\S_5\end{bmatrix}=\begin{array}{c}j\\ \begin{bmatrix}\cdots&0&\cdots\\ \cdots&1&\cdots\\ \cdots&0&\cdots\\ \cdots&1&\cdots\\ \cdots&0&\cdots\end{bmatrix}\end{array}$$

(b)

Figure 4 An illustrative example of the generated control sequences at a time step for a network of five genes. (a) The control sequences for the randomly one gene (ROG) model. (b) The control sequences for the randomly m genes (RMG) model. S_i indicates the control sequence for gene i, i∈[1,5]; j is for the jth bit in each sequence, which can be considered as the jth trial of the L experiments and L indicates the length of the stochastic sequences and j∈{1,2,⋯,L}. S_(i,j)=0 indicates that the control bit is 0 for the asynchronous update module (AUM) in Figure 3, so the state of gene i remains, while S_(i,j)=1 means that the next state is determined by the predictor function F_i. For the RMG model, the number of genes to be updated for the jth trail is assumed to be two, thus ∑_(i=1)^n▒S_(i,j) =2 in (b). n is the number of genes in the investigated network.
56x23mm (300 x 300 DPI)

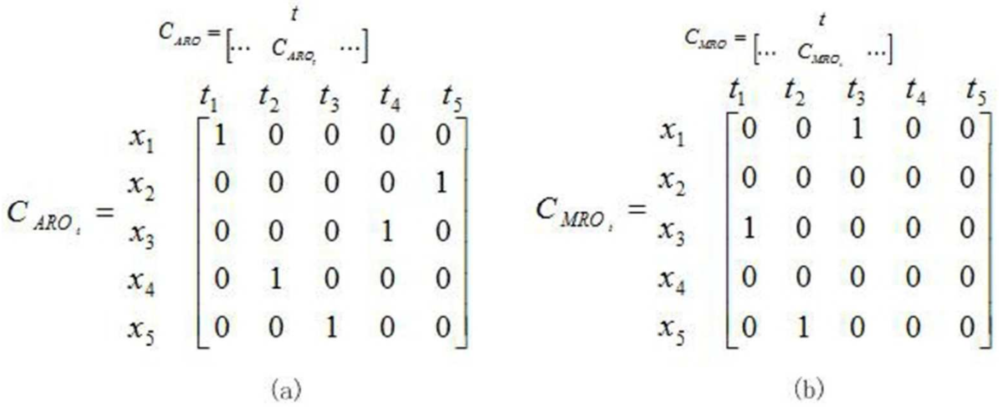
$C_{ARO} = [\ldots \overset{t}{C_{ARO_t}} \ldots]$
$C_{ARO_t} = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ x_1 & 1 & 0 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 & 0 & 1 \\ x_3 & 0 & 0 & 0 & 1 & 0 \\ x_4 & 0 & 1 & 0 & 0 & 0 \\ x_5 & 0 & 0 & 1 & 0 & 0 \end{array}$
(a)
$C_{MRO} = [\ldots \overset{t}{C_{MRO_t}} \ldots]$
$C_{MRO_t} = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ x_1 & 0 & 0 & 1 & 0 & 0 \\ x_2 & 0 & 0 & 0 & 0 & 0 \\ x_3 & 1 & 0 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 0 & 0 & 0 \\ x_5 & 0 & 1 & 0 & 0 & 0 \end{array}$
(b)

Figure 5 An illustrative example of the generated control sequences at a time step for a network of five genes. (a) The control sequences for the all genes updated in a random order (ARO) model. (b) The control sequences for the m genes updated in a random order (MRO) model. t is for the tth time step and t_1,⋯,t_5 are the sub-steps at time t. 〖C_ARO〗_t and 〖C_MRO〗_t are the generated control sequences at the sub-steps of time t. An entry of 0 indicates the state of gene i remains while a value of 1 indicates that the next state is determined by the predictor functions. The number of genes to be updated at time t is three (in a random order), thus ∑▒〖C_MRO〗_t =3 in (b).
66x27mm (300 x 300 DPI)

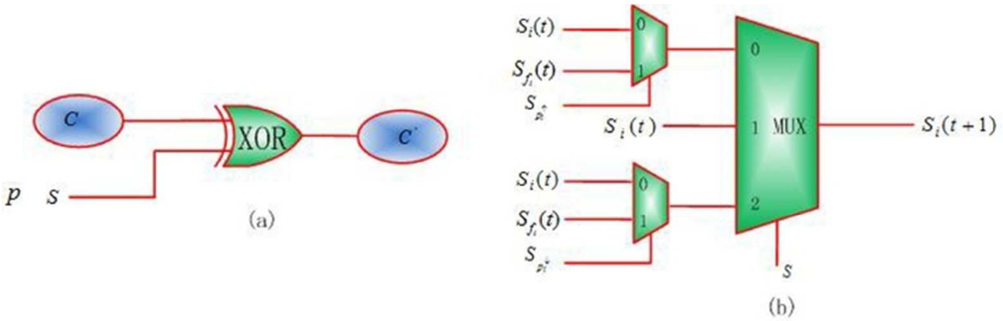

Figure 6 Stochastic architectures for the models of stochasticity in node (SIN) and stochasticity with propensity parameter (SPP). (a) The stochastic architecture of the SIN model for a binary gene node [12]. (b) The stochastic architecture of the SPP model for updating the state of a gene.
60x19mm (300 x 300 DPI)

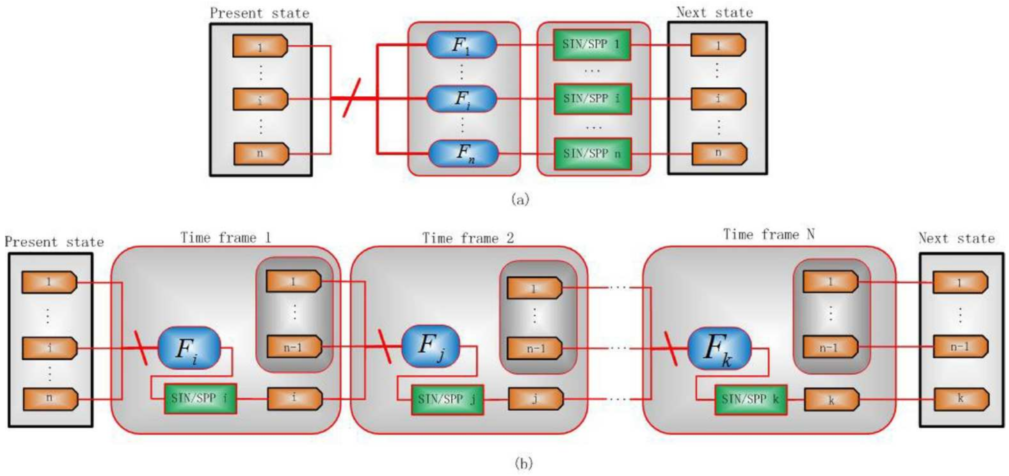

Figure 7 Synchronous and asynchronous stochastic Boolean networks (ASBNs) for different stochasticity models. (a) A synchronous SBN. (b) An ASBN for the randomly one gene (ROG) model. F_i: predictor function for gene i, i∈{1,2,⋯,n}. n is the number of genes in the investigated network. The detailed structures and inputs of the SIN and SPP modules are shown in Figure 6 (a) and (b) respectively.
130x61mm (300 x 300 DPI)

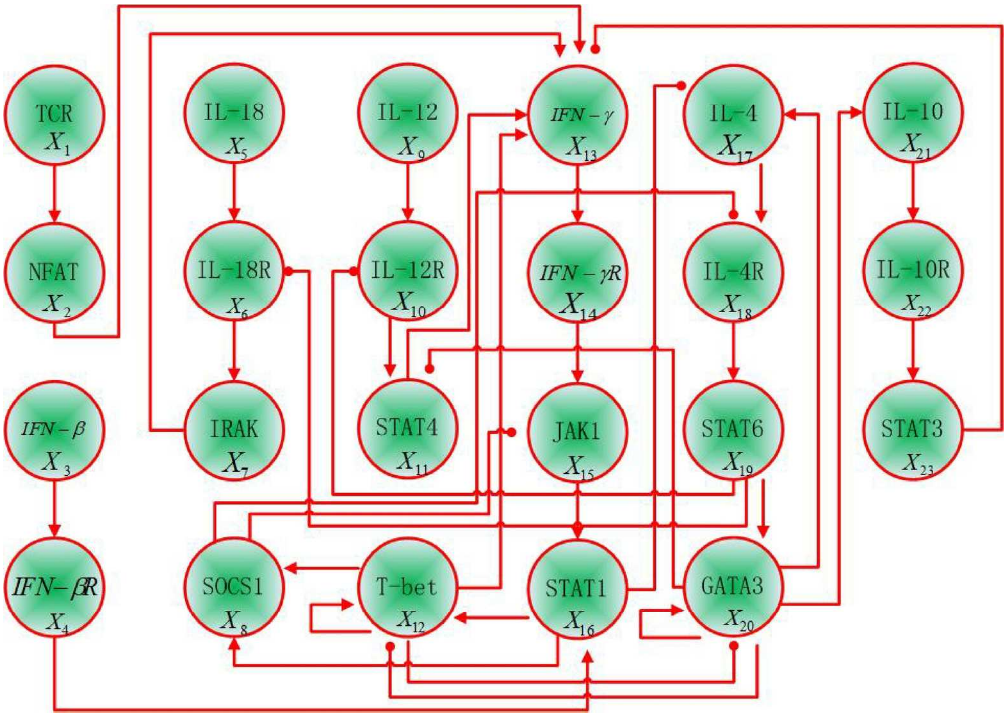

Figure 8 A T helper network. A positive regulatory interaction is indicated by a regular arrow, while a negative interaction is represented by a blunted arrow [31].
179x127mm (300 x 300 DPI)

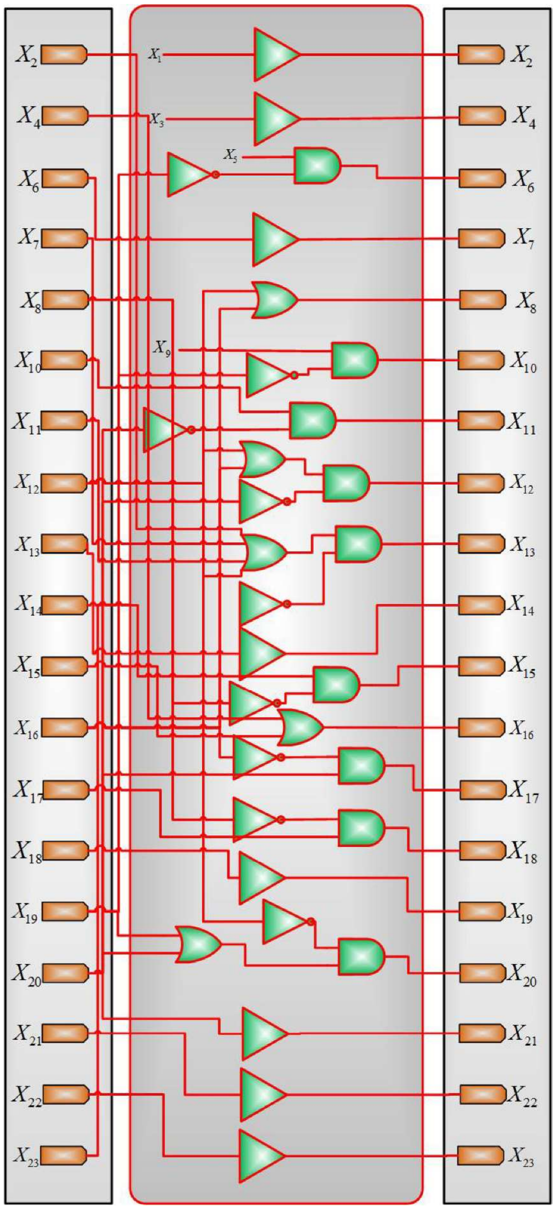

Figure 9 A deterministic BN model for the 23-gene T helper network, with X_1=0, X_3=0, X_5=0 and X_9=0. X_i, i∈{1,2,⋯,23}, indicates a gene node in Figure 8.
392x857mm (300 x 300 DPI)

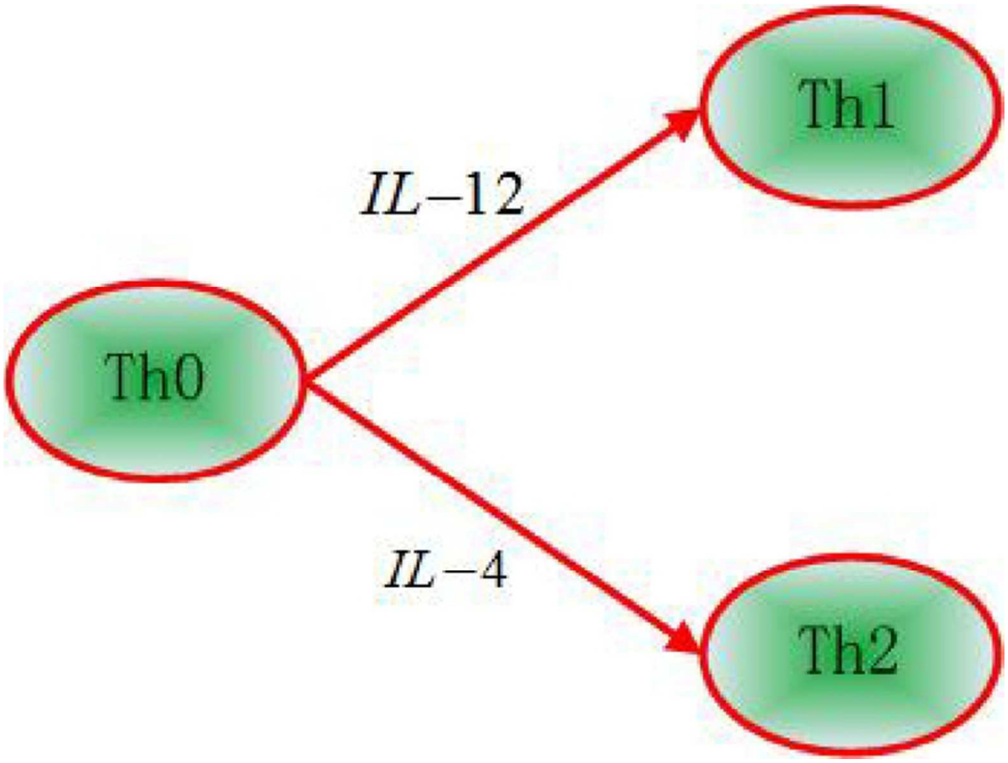

Figure 10 Differentiation of T helper (adapted from [33]). Th1 cells express IFN-γ and T-bet, while Th2 expresses GATA3 and IL-4.
69x52mm (300 x 300 DPI)

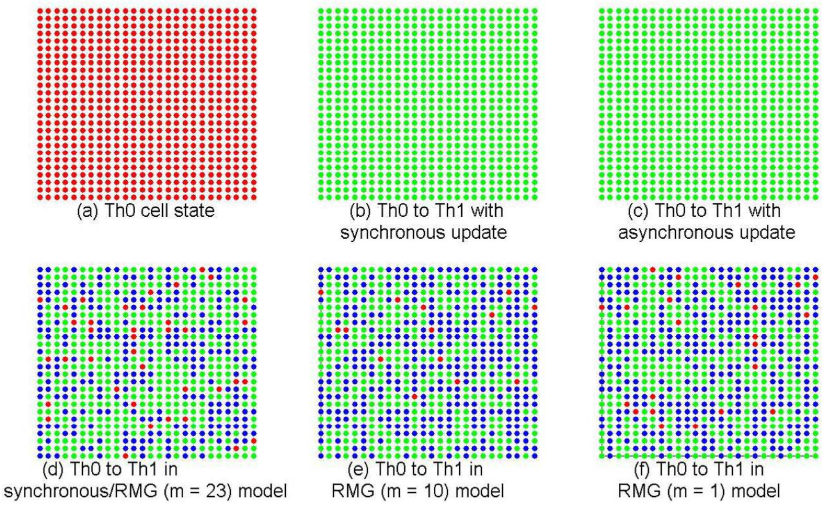

Figure 11 Transitions between attractors during the differentiation process of the T helper network with an external stimulus of IL-12 for the synchronous and randomly m genes (RMG) models. Each dot indicates a T helper cell and each cell is independent of the neighboring cells. A red dot represents an undifferentiated cell at Th0, a green dot indicates the cell state of Th1 and a blue dot indicates the state of Th2. (a) All cells are in the initial Th0 state. (b) and (c) The transitions from Th0 to Th1 by synchronous and asynchronous updating rules (without perturbation to the network). (d), (e) and (f) The transitions from Th0 for the stochasticity in node (SIN) model with perturbation probability of 0.5 for each node: the percentage of the corresponding nodes indicates the percentage of the cells deriving into an attractor. For the first five time frames, each of the gene nodes is subject to the effect of SIN. The obtained transition probabilities from Th0 to different attractors are similar to the values in Table 1, because it is equivalent to setting the initial state as the wile type by having each of the nodes under SIN for several time frames.
284x155mm (300 x 300 DPI)