IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 9, 2021, accepted December 30, 2021, date of publication January 6, 2022, date of current version January 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140720

Density Evolution for Noise Propagation Analysis in Biological Networks

STEPHEN KOTIANG, (Member, IEEE), AND ALI ESLAMI, (Member, IEEE)
Department of Electrical and Computer Engineering, Wichita State University, Wichita, KS 67260, USA

Corresponding author: Ali Eslami (ali.eslami@wichita.edu)

This work was supported in part by the National Aeronautics and Space Administration (NASA) under Award 80NSSC20M0133.

ABSTRACT Accurate prediction of noise propagation in biological networks is key to understanding faithful signal propagation in gene networks as well as for designing noise-tolerant synthetic gene circuits. Knowledge on how biological fluctuations propagate up the development ladder of biological systems is currently lacking. Similarly, little research effort has been devoted to the analysis of error propagation in biological networks. To capture and characterize error evolution, this paper considers a Boolean network (BN) model representation of a biological network such that nodes on the graph represent diverse biological entities, e.g., proteins, genes, messenger-RNAs, etc. In addition, the network edges capture the interactions between nodes. By conducting a density evolution analysis on the graphical model based on node functionalities, a recursive closed-form expression for error propagation is derived. Subsequently, the recursive equation allows us to obtain a necessary condition to guarantee noise-error elimination in dynamic discrete gene networks. Our analytical formulations provide a step toward achieving optimal network parameters for resilience against variability or noise in biology.

INDEX TERMS Boolean networks, density evolution, error propagation, factor graph, gene regulatory networks.

I. INTRODUCTION

The execution of biological functions relies on faithful signal propagation from one gene to another. The execution process may be hindered by biological fluctuations that introduce noise or variability in gene expressions. In the literature, noise is considered a fundamental, inherent aspect of gene expression, having a diverse functional role in biological processes [1]. This noise can emanate from either environmental or biological fluctuations due to small changes in external stimuli such as pH changes, mutagens, heat stress, etc. [2]. According to Wang and Zhang [3], variability or randomness in gene expression more often presents a hindrance to proper biological functions such as decision-making, development, spatiotemporal population dynamics, and reaching optimal fitness. In addition, it poses entropy-increasing effects of limiting signal fidelity, robustness, and channel capacity of signaling relays. For example, in genomic signal processing, genetic switches that control cellular decisions can flip under fluctuations that bring the biological system close to the threshold for a phase transition [1].

Despite the hindrances introduced by gene expression variation, studies indicate that fluctuations can be biologically meaningful [1], [4]–[6]. Variability plays an important role in natural resistance to harmful chemicals [5], and has been found useful for balancing precision and diversity in eukaryotic gene expression [6]. In addition, stochasticity in biology becomes a benefit when it is used to generate novelty at higher levels of organization [4], [7].

The majority of studies on biological noise are concerned mainly with elucidating sources of gene expression noise [2], understanding mechanisms and effects of expression variations in biological systems [8]–[13], and characterizing noise properties using small-noise approximations [14], [15]. However, little research emphasis has been placed on the study and quantification of noise propagation in biological systems. Noise propagation is considered important in assessing the information capacity of a regulatory interaction, i.e., the number of distinct stable states of a target gene expression level that can be achieved by varying the concentration of a transcription factor [16]. Knowledge of noise propagation can

The associate editor coordinating the review of this manuscript and approving it for publication was Yichuan Jiang.

enable us to develop strategies that enhance the performance of downstream cascades in many biological applications. For instance, to engineer predictable behavior through synthetic gene networks, one must develop reliable means for connecting smaller functional artificial gene circuits to realize predictable high-order networks.

Pedraza and van Oudenaarden [17] quantified how noise propagates in gene networks by measuring expression correlations between genes in single cells. They considered a simple synthetic gene network consisting of only four genes. Their approach provides a step toward understanding noise propagation; however, no rigorous analysis or how the method can be extended into more complex biological networks is provided. In another work, Arola-Fernández *et al.* [18], applied error propagation to estimate uncertainty in the critical threshold for some dynamical processes in complex networks with noisy links. In their work, they considered network noise resulting from experimental errors such as device accuracy, sampling biases, or data entry. However, they did not analyze errors emanating from gene expression variations.

In biological networks, feedback loops are common and are known to have a critical role in cellular signaling [19], [20]. Zhang *et al.* [19] employed frequency domain analysis [20], [21] to investigate the role of feedback loops in sensitivity and noise amplification on the dynamical behavior of a biological system. They observed that interlinked positive and negative feedback loops dynamically tune noise propagation signals rather than monotonically suppressing or amplifying them, as would be expected in single feedback loops. However, their biochemical and kinetic network model demands more knowledge, namely kinetics of individual processes as well as deduction of signal sensitivity and noise-amplification parameters. Information on kinetic parameters can be extracted from the literature; however, this is difficult to find [22].

In this paper, we study and quantify noise propagation in regulatory networks of gene interactions by applying a messages-passing algorithm that evolves network states. For analytical purposes, we consider noise to introduce errors randomly on the state of nodes in the network. We explore Boolean network (BN) model representation of biological networks where nodes represent biological entities such as proteins, mRNAs, genes, etc., and the edges represent the types of interaction, either inhibiting or activating, between nodes [23], [24]. BNs are simple and have proven to effectively and qualitatively explain some fundamental characteristics, e.g., steady-state behavior, of biological networks. To analyze error propagation in a gene network, we transform the BN model into an equivalent factor graph or bipartite graph following the formulation introduced in [25]. Subsequently, we apply a message-passing algorithm as an inference technique to the factor graph model, where messages represent the states of the nodes plus the error function in the network.

The factor graph representation is convenient and is known for its many applications in probability theory, systems biology, cyber-physical systems, coding theory, and complex networks [25]–[28]. In systems biology, the factor graph representation has been successfully employed to predict the impact of single gene deletions on the global dynamic behavior of gene regulatory networks (GRNs) [25], refine regulatory networks of gene interactions with improved fit to experimental data and discover new regulatory relationships [27], optimize gene regulation functions [29], and even characterize the steady-state behavior of Bayesian gene networks [30]. We employ the concept of *density evolution* (DE) used in the performance analysis of factor graphs [25], [26] to characterize error propagation in biological networks after an initial disturbance. The resultant closed-form recursive formula derived is referred to as the ''density evolution equation.'' Based on the DE equation, we derive a necessary condition for network parameters to guarantee vanishing errors or optimize the design of a biological system for high signal fidelity.

In [25], we introduced density evolution as a proof of concept in the analysis of error propagation in biological networks. We considered the activation-inhibition Boolean functions that encode majority voting rule to determine the next state of a node in the network. The majority rule is simple; however, it is less likely used in many biological models that often rely on complex logical rules to account for biological processes such as cooperative effects, selective inhibitions, and dual interactions [23], [31].

The rest of this paper is organized as follows: In section II, we briefly review background material regarding the use of factor graphs, message passing, and density evolution in graphical models. Then, Section II-C starts the error propagation analysis by applying density evolution to gene regulatory networks. Section III is devoted to discussion, results, and performance analysis. Finally Section IV concludes the paper.

II. MODEL AND METHODS

In this section, we provide background information on factor graphs and the message-passing inference algorithm, which is required to understand the analysis described here. We explain the general message-passing technique, followed by a brief description of density evolution analysis employed on factor graphs. Then, we introduce DE in the analysis of GRNs.

A. FACTOR GRAPHS AND MESSAGE-PASSING ALGORITHM

In general, a factor graph for variables $x_1, \cdots, x_n$ and functions $f_1, \cdots, f_K$ could be defined as a bipartite graph associating a set of nodes known as *variable nodes* corresponding to the variables, and a set of *control nodes* corresponding to the functions [32]. Each control node depends on a subset of variable nodes, i.e., an edge exists between variable node x_i and control node f_k, if and only if x_i is an argument of f_k. The

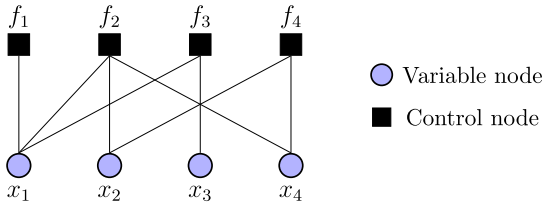

FIGURE 1. Factor graph for joint distribution product $f_1(x_1)f_2(x_1, x_2, x_4)f_3(x_1, x_3)f_4(x_2, x_4)$.

joint distribution function over the variables is given by

$$p(x_1, \cdots, x_n) = \prod_{k \in K} f_k\left(x_{C_k}\right), \tag{1}$$

where K is a discrete index set, C_k is the index set of variables that are connected to the function f_k, and x_{C_k} denotes this set of variables. Also, we refer to each control $f_k\left(x_{C_k}\right)$ in (1) as a *local function*. Fig. 1 shows an example of a function $p(x_1, x_2, x_3, x_4)$ on a bipartite graph that can be obtained as the product of $f_1(x_1)f_2(x_1, x_2, x_4)f_3(x_1, x_3)f_4(x_2, x_4)$.

Factor graphs encompass many graphical models including Markov random fields [33], Bayesian networks [34], and Tanner graphs [28], [35]. It is plausible that many algorithms and mathematical models in these fields are naturally expressed in terms of factor graphs. One such algorithm is the *sum-product algorithm*, also known as *belief-propagation* [32], which operates in a factor graph by passing "messages" along edges of the graph, following a single, simple computational rule.

In many circumstances, we seek to compute the *posterior* distributions, also referred to as marginal functions, $p_i(x_i)$, for more than one value of i. In the sum-product message-passing algorithm, it is worth noting that there is no fixed parent/child relationship among neighboring nodes. As such, each two neighboring nodes are regarded as a parent of the other at some point. The computation of $p_i(x_i)$ begins at the leaves (i.e., nodes without descendants) of a factor graph. Each leaf control node and each leaf variable node send "belief" messages to their parent nodes. A control node f with parent x computes the product of all received messages from its children and then operates on the result with a summary operator $\sum_{\sim\{x\}}$ over all its variables except x to send its belief. On the other hand, a variable node x simply sends the product of all received messages as its belief.

To obtain a mathematical expression for the belief-propagation, let $n(v)$ denote a set of all neighbors of a node v, $\mu_{x\to f}(x)$ represent a message sent from node x to f, and let $\mu_{f\to x}(x)$ denote the message sent from control node f to variable node x. Therefore, as illustrated in Fig. 2, the message computations are expressed as follows [32]:

$$\mu_{x\to f}(x) = \prod_{k \in n(x)\setminus\{f\}} \mu_{k\to x}(x), \tag{2}$$

$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left(f\left(x_C\right) \prod_{y \in n(f)\setminus\{x\}} \mu_{y\to f}(y) \right), \tag{3}$$

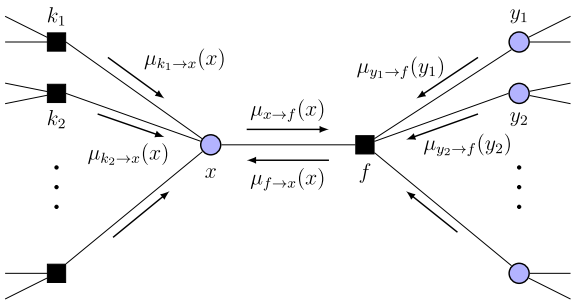

FIGURE 2. Factor-graph fragment depicting update rules of sum-product message-passing algorithm.

TABLE 1. Symbols used in the model and analysis.

Symbol	Description
$\mu_{x\to f}(x)$	message sent from variable node x to control node f
$\mu_{f\to x}(x)$	message sent from control node f to variable node x
x_a	logical combination of activators
x_r	logical combination of repressors
ϵ_x	probability of error on node x
ϵ_l	average propagated error in l-th iteration
k_a (k_r)	number of activating (inhibiting) edges into node
ρ_{ij}	fraction of edges incident to control node with degree i activators and degree j repressors

where $x_C = n(f)$ is the set of arguments of the function f. Table 1 lists a summary of symbols and their definitions as used in this article.

Considering $p(x_3)$ in Fig. 1 as an example, $p(x_3)$ can be obtained as follows:

$$p(x_3) = \mu_{f_3\to x_3}(x_3), \tag{4}$$

where

$$\mu_{f_3\to x_3}(x_3) = \sum_{\{x_1\}} f_3\left(x_1, x_3\right)\ \mu_{x_1\to f_3}(x_3), \tag{5}$$

and

$$\mu_{x_1\to f_3}(x_3) = \mu_{f_1\to x_1}(x_1)\ \mu_{f_2\to x_1}(x_1). \tag{6}$$

At the end of each message-passing cycle, a variable node updates according to the rule given in (2), while the update rule at a local function node is given in (3). This iterative process continues until convergence, i.e., when no significant difference in belief update occurs. Then the algorithm terminates, and we compute $p_i(x_i)$ as the product of all messages directed toward x_i.

B. DENSITY EVOLUTION ON FACTOR GRAPHS

In the literature, density evolution refers to tracking the probability density function (pdf) of error messages between variable nodes and control nodes with the aim of studying the behavior and performance of a factor graph model [28], [36]. As an illustrative example, consider the factor graph shown

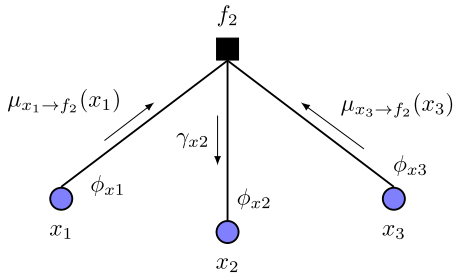

FIGURE 3. Message-passing in factor graph with erroneous messages. $\phi_{(\cdot)}$ represents probability of error message from variable to control node, whereas $\gamma_{(\cdot)}$ denotes probability of error message from control to variable node.

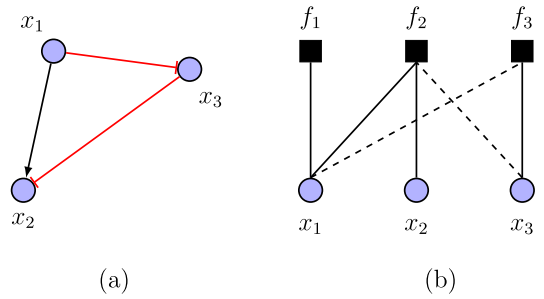

FIGURE 4. (a) Simple directed gene graph, and (b) equivalent undirected factor graph representation of (a). Red blunt edges in (a) and black dashed edges in (b) indicate inhibition links, whereas black arrowheads represent activation interactions.

in Fig. 3 having a modulo 2 check-sum control node functionality on the values of its neighboring variable nodes to zero. A belief-propagation algorithm is applied to the graph, where the messages passed between the variable and control nodes are defined as log-likelihood ratios of probabilities that a given state is '0' or '1'. Let ϕ_l denote the probability of an error message from a variable node to a control node, and let γ_l represent the probability of an error from a control node to a variable node, both in the l-th iteration of a message-passing algorithm. Thus, conditioned on the event that the control node has degree d, we have $\gamma_l = 1-(1-\phi_l)^{d-1}$ under the assumption of independence [28]. Of note, the degree of a node in the factor graph denotes the number of edges incident to the node.

To obtain a closed-form expression for the total error probability in the graph, from control nodes to variable nodes, we define a generating function $\rho(z) = \sum_d \rho_d z^{d-1}$, which represents the control node degree distribution, where ρ_d is the fraction of nodes with degree d. Hence, we obtain the following: $\gamma_l = 1 - \rho(1 - \phi_l)$. For this example, we assume that a variable node performs no computation on the received messages but simply sends back the received messages on each edge. Then, for the next iteration, $\phi_{l+1} = \gamma_l$. Hence, we can write the closed form recursive expression for the graph as $\phi_{l+1} = 1 - \rho(1 - \phi_l)$.

In [25], we hypothesized that density evolution can be employed to provide an exact analytic characterization of the impact of random network perturbation on the steady-state behavior of a biological network. Additionally, DE analysis has also been employed in other research fields such as multi-layered complex networks and coding theory. The authors in [26] used DE analysis to study the dynamics of failure propagation and healing in networked cyber-physical systems. Likewise, in [28], density evolution was employed to track the density of erasure messages in order to analyze the performance of a decoding algorithm for low-density parity-check codes.

C. DENSITY EVOLUTION ANALYSIS OF GRNs

This section presents a Boolean network model and its equivalent factor graph model in biological systems. Given an initial disturbance due to noise in the system, we then derive a recursive DE equation that numerically characterizes network parameters for resiliency and robustness against network disturbance.

1) NETWORK MODEL

For our analysis, let us consider a Boolean gene regulatory graph defined by a set of n binary-valued nodes $\{x_1, \cdots, x_n\}$ representing biological entities, and a list of Boolean functions $\{f_1, \cdots, f_n\}$ denoting the rules of regulatory interaction between the nodes (i.e., genes). Each $x_i \in \{0, 1\}$, where $i = 1, \cdots, n$ has k_i nodes assigned to it, which completely determines its value at time $t + 1$ following f_i, that is,

$$x_i(t+1) = f_i\Big(x_{i1}(t), x_{i2}(t), \cdots, x_{ik_i}(t)\Big), \tag{7}$$

where $\{i1, \cdots, ik_i\}$ is the index set of variables that are connected to node x_i. The state of x_i denotes the expression of the gene, where $x_i = 1$ indicates that the gene is active (expressed), and $x_i = 0$ means the gene is inactive. Here, we assume all genes update synchronously in accordance with the f_i's assigned to them, and the process is repeated iteratively in time. In addition, in this paper, we consider the activation-inhibition Boolean functions proposed in the literature [23], [31] that encode complex biological rules or processes such as the cooperative effect of the interacting genes of the form $x(t+1) = \big(x_{a1}(t) \vee x_{a2}(t) \vee \cdots\big) \wedge \neg\big(x_{r1}(t) \vee x_{r2}(t) \vee \cdots\big)$, where $x_{a1}, x_{a2}, \cdots$ are activators, and $x_{r1}, x_{r2}, \cdots$ are inhibitors or repressors acting on the node. The logical operators $\{\vee, \wedge, \text{and } \neg\}$ bear the usual meanings. An example of a gene network with three genes and its equivalent bipartite graph is shown in Fig. 4. The state of a gene x_i at time $t+1$, for instance, $x_2(t+1) = f_2\big(x_1(t), x_3(t)\big)$, is given by $x_1(t) \wedge \neg\, x_3(t)$.

In this work, to evolve network states on the factor graph, a control node receives input from its neighboring variable nodes, computes an output according to (7), and passes it to its corresponding variable node. In the subsequent iteration, a variable node simply sends its current state to its neighboring control nodes, i.e., a variable node performs no computation on its received message. The network states evolve iteratively until a stable state is attained.

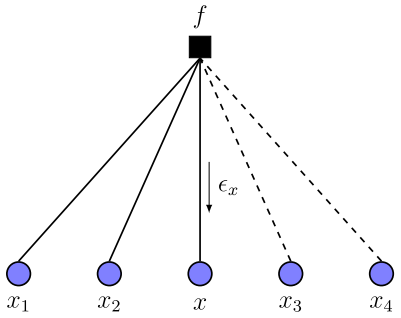

FIGURE 5. Factor graph with initial disturbance error probability ϵ. Genes x_1 and x_2 activate gene x, whereas genes x_3 and x_4 inhibit gene x. Dashed edges denote inhibition interactions.

TABLE 2. Boolean truth tables for implementing logical cooperative rule in (8).

Activation		
x_1	x_2	x_a
0	0	0
0	1	1
1	0	1
1	1	1

Repression		
x_3	x_4	x_r
0	0	1
0	1	0
1	0	0
1	1	0

Cooperative Effect		
x_a	x_r	$x_a \wedge x_r$
0	0	0
0	1	0
1	0	0
1	1	1

2) ERROR ANALYSIS

Consider a portion of a factor graph with five variable nodes, as shown in Fig. 5. Nodes x_1 and x_2 activate node x, whereas nodes x_3 and x_4 repress x. We have implemented a logical cooperative effect at the control nodes following the activation-inhibition function proposed in [23]. The next state of node x is given by the formulation

$$x = \underbrace{(x_1 \vee x_2)}_{x_a} \wedge \underbrace{\neg(x_3 \vee x_4)}_{x_r}, \tag{8}$$

where x_a (x_r) is the logical combination of all activators (repressors) regulating a node, respectively. Therefore, $x = x_a \wedge x_r$. Each node in the graph is initially disturbed with a small probability ϵ, where $\epsilon \ll 1$, independently from other nodes. We assume that ϵ is sufficient enough to change the state of a node. Let ϵ_x denote the event that there is an error on node x due to random error from its regulators. To study the overall error propagation in the biological network, we represent the logical cooperative effect using truth tables, as shown in Table 2.

Furthermore, let ϵ_l represent the propagated error in the l-th iteration of message passing. We may look at the evolution within one (any) iteration, and see how the disturbance probability changes for the gene nodes. If this probability increases at the end of each iteration, then a cascade will occur. However, if it decreases, then the effect of disturbance is minimized. Let us consider the l-th iteration on node x in Fig. 5. Using the cooperative effect truth table in Table 2, we can represent the probability of error in node x as

$$\begin{aligned}
\epsilon_l &= \Pr(\epsilon_x | x_a = 0, x_r = 0) \cdot \Pr(x_a = 0, x_r = 0) && \text{(9a)}\\
&+ \Pr(\epsilon_x | x_a = 1, x_r = 0) \cdot \Pr(x_a = 1, x_r = 0) && \text{(9b)}\\
&+ \Pr(\epsilon_x | x_a = 0, x_r = 1) \cdot \Pr(x_a = 0, x_r = 1) && \text{(9c)}\\
&+ \Pr(\epsilon_x | x_a = 1, x_r = 1) \cdot \Pr(x_a = 1, x_r = 1). && \text{(9d)}
\end{aligned}$$

We obtained (9) by considering the probability of having an error on node x, i.e., ϵ_x, given the input states, x_a and x_r. Assuming that we are at the beginning of the l-th iteration, we describe this error in terms of the $(l-1)$-th iteration, i.e., $\epsilon_l = f(\epsilon_{l-1})$. The detailed derivation to solve (9) is shown in the Appendix. It yields

$$\begin{aligned}
\epsilon_l = \frac{1}{2^{2k_i}} \Big\{ & 2^{k_i} \left(2^{k_a} - 1\right) - \Big[\left(2^{k_a+1} - 1\right) (1-\epsilon_{l-1})^{k_a} \\
& - 3 \cdot 2^{k_a} + 4^{k_a} + 1 \Big] (2 - 2\epsilon_{l-1})^{k_r} \\
& - \left(2^{k_r} - 1\right) \left[1 - (1-\epsilon_{l-1})^{k_a} - 2^{k_a} + 4^{k_a}\right] \\
& \times \left[(1-\epsilon_{l-1})^{k_r} - 1 \right] \Big\},
\end{aligned} \tag{10}$$

where k_a (k_r) denotes the cardinality of activating (inhibiting) edges of a node, respectively. In addition, $k_i = k_a + k_r$ is the discrete index set referred to as the degree of a node x_i.

To account for the average error probability in the network, we employ polynomials that represent the degree distributions of the networks in terms of activation and inhibition as follows:

$$\rho(u, v) = \sum_{i,j \geq 0} \rho_{ij}\, u^i v^j, \tag{11}$$

where ρ_{ij} is the fraction of edges incident to a control node with degree i activators (u) and degree j repressors (v), constrained to $\sum_{i,j\geq 0} \rho_{ij} = 1$. The polynomial representation allows us to model random networks with arbitrary degree distributions such as biological networks. Of note, $\rho(u, v)$ is obtained from the topology of the network. Therefore, given a biological network with control node degree distribution $\rho(u, v)$ and experiencing a perturbation, the average error probability of the network can be obtained as

$$\begin{aligned}
\epsilon_l = \sum_{k_a, k_r \geq 0} \frac{\rho_{k_a k_r}}{2^{2k_i}} \cdot \Big\{ & 2^{k_i} \left(2^{k_a} - 1\right) - \Big[\left(2^{k_a+1} - 1\right) \\
& \times (1-\epsilon_{l-1})^{k_a} - 3 \cdot 2^{k_a} + 4^{k_a} + 1 \Big] (2 - 2\epsilon_{l-1})^{k_r} \\
& - \left(2^{k_r} - 1\right) \left[1 - (1-\epsilon_{l-1})^{k_a} - 2^{k_a} + 4^{k_a}\right] \\
& \times \left[(1-\epsilon_{l-1})^{k_r} - 1 \right] \Big\}.
\end{aligned} \tag{12}$$

Equation (12) represents the recursive DE equation for the cooperative effect biological model in (8). After an initial perturbation occurs in the network, error messages appear in the network. It is expected that the impact of disturbance in the network diminishes if and only if $\epsilon_l \to 0$ as $l \to \infty$.

Once the recursive equation of density evolution is obtained for the network, it can be used to gain useful insights into the structure of biological networks and allow us to quantify network parameters in designing models for reverse engineering gene networks from gene-expression profiles. Consider the recursive DE equation (12). By taking the Taylor series from the right-hand side of (12) at $\epsilon_{l-1} = 0$, we obtain

$$\epsilon_l = \beta_{u,v}\epsilon_{l-1} + O(\epsilon_{l-1}^2), \tag{13}$$

where

$$\beta_{u,v} = \Big[\frac{3}{4}\rho_{10} + \frac{3}{4}\rho_{11} + \frac{5}{8}\rho_{12} + \frac{57}{128}\rho_{13} + \frac{7}{8}\rho_{20} + \rho_{21} + \frac{7}{8}\rho_{22} + \frac{163}{256}\rho_{23} + \frac{45}{64}\rho_{30} + \frac{129}{128}\rho_{31} + \frac{241}{256}\rho_{32} + \frac{45}{64}\rho_{33}\Big]. \tag{14}$$

In this series expansion, we have computed ϵ_l for every $\{k_a, k_r\} = \{0, \cdots, 3\}$. Here, we limit the maximum value of $\{k_a, k_r\} = 3$. Similarly, in most gene network reconstruction tools and algorithms proposed in the literature, researchers limit the size of each node's parents. This is done to reduce the computational complexity required to search all combinations of parents [37]–[39]. For vanishing errors, i.e., ϵ_l to be less than ϵ_{l-1}, we conclude from (13) that

$$\beta_{u,v}\epsilon_{l-1} < \epsilon_{l-1}. \tag{15}$$

Inequality (15) holds for every l if it holds for $l = 1$. Therefore, the necessary condition becomes $\beta_{u,v} < 1$. From (14), we note that ρ_{21} and ρ_{31} are the dominant factors having coefficients ≥ 1 . Also, factors ρ_{01}, ρ_{02}, and ρ_{03} are absent; thus, we hypothesize that nodes with only inhibiting regulators have less significance in cascading errors in biological systems.

III. DISCUSSION AND RESULTS

In this section, we provide a quantitative evaluation of the proposed error propagation model in MATLAB environment. We evaluate the density evolution equation (12) and the inequality $\beta_{u,v} < 1$, and then provide a computational complexity analysis of the propagation model.

A. NUMERICAL EVALUATIONS

First, we evaluate (12) for a sample random Boolean network with parameter value $\beta_{u,v} = 0.6651$, where the maximum k_a and k_r value is 3. We exemplify a random network with random $\rho_{k_a k_r}$ values such that $\sum_{k_a,k_r\geq 0}^{3} \rho_{k_a k_r} = 1$, resulting in a $\beta_{u,v}$ value of 0.6651 following (14). In Fig. 6, we show the total propagated error for different numbers of iterations, l. At the end of the first iteration, the average propagated error

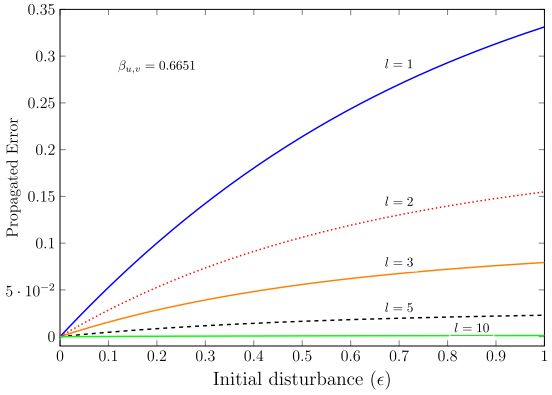

FIGURE 6. Propagated error in biological network with $\beta_{u,v} = 0.6651$ in different iterations w.r.t. the initial disturbance. As $l \to \infty$, the evolved error vanishes in $\beta_{u,v} < 1$.

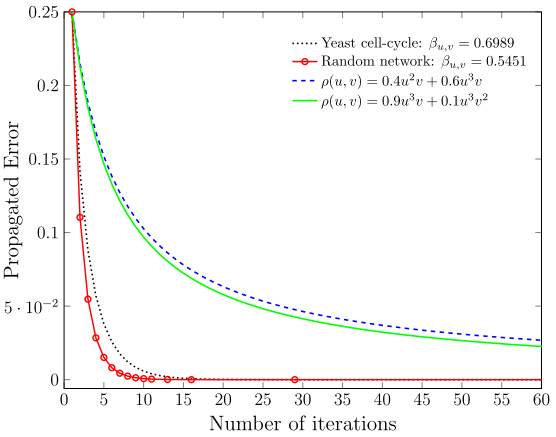

FIGURE 7. Probability of error for networks with different polynomial degree distributions, i.e., $\beta_{u,v}$ values (shown in the plot) versus the number of iterations, $l \to \infty$. Initial disturbance $\epsilon_0 = 0.25$. Networks $\rho(u,v) = 0.9u^3v + 0.1u^3v^2$ and $\rho(u,v) = 0.4u^2v + 0.6u^3v$ both have $\beta_{u,v} > 1$.

on any particular node decreases, and so $\epsilon_l < \epsilon_{l-1}$ if and only if $\beta_{u,v} < 1$.

Similar to observations made in decoding algorithms for codes on graph, the network or graph structure plays an important role in the propagation of errors or noise in genetic graphs as depicted in Fig. 7. In Fig. 7, the plots show the quantity of error propagated in sample Boolean gene graphs, such as the budding yeast cell-cycle network model presented by Li *et al.* [40], a random network with $\beta_{u,v} = 0.5451$, and two theoretical networks represented using polynomial degree distributions given by $\rho(u,v) = 0.9u^3v + 0.1u^3v^2$ and $\rho(u,v) = 0.4u^2v + 0.6u^3v$, both of which guarantee that $\beta_{u,v} > 1$. Moreover, the yeast logical network [40] considered has the following polynomial degree distribution according to (11):

$$\rho(u,v) = 0.0345u + 0.1379uv + 0.1034uv^2 + 0.1379uv^3 + 0.1379u^2 + 0.1034u^2v + 0.3448u^2v^3. \tag{16}$$

The polynomial degree distribution (16) yields $\beta_{u,v} = 0.6989$, which implies that the network may be resilient against random state disturbances. As can be seen in Fig. 7, for networks with $\beta_{u,v} < 1$, an error introduced due to a network disturbance vanishes as $l \to \infty$, in accordance with the inequality (15). In addition, a violation of this inequality is such that $\epsilon_l \not\to 0$. For instance, the two networks evaluated in Fig. 7 with dominant ρ_{21} and ρ_{31} factors are such that $\beta_{u,v} > 1$; thus, their corresponding line plots do not converge to zero. If the error vanishes, then we may deduce that the initial states of the network are reset, and therefore, the steady-state distribution of the network remains invariant. Conversely, a non-zero error indicates that the initial perturbation throws the network out of its steady-state condition, which is consistent with several theoretical and experimental studies [10], [20]. That is, gene expression noise can create new stable states or destabilize existing ones [11]–[13].

Of note, in [40], the authors observed that the yeast logical network has an observable large stable state, thus making it robust against network perturbations. Our results highlight the importance of degree distribution restrictions on genetic graphs that will be necessary for understanding signal fidelity in natural networks as well as in the design of noise-tolerant artificial gene circuits. This observation supports conclusions made in the literature that the topology of a genetic graph does not only determine its information-procession capability but also encodes its sensitivity to noise [10], [20]. In simple terms, network topology provides a means to locally tune noise propagation.

Here, we have assumed that the error introduced into the network is sufficient to change the state of nodes randomly in the graph. Unlike the check node functionality in codes on graphs that is simple and only performs a modulo 2 operation to derive what it believes about the value of its neighboring nodes, the control nodes in a biological factor graph may assume a more complicated functionality. For example, to a certain extent, the cooperative effect biological process implemented in this work using discrete states, tends to correct errors introduced in the network. From the DE equation, we obtained the necessary condition as a function of the network connectivity, i.e., $\beta_{x,y} = f(\rho_{ij})$. Hence, we deduce that network structure plays a critical role in the stability of biological graphs.

B. PERFORMANCE ANALYSIS

1) COMPUTATIONAL COMPLEXITY

In our proposed model, the overall computational cost of characterizing the propagated error in the network is given by the complexity of the message-passing algorithm employed on the factor graph model. This cost grows linearly with the number of edges in the network [25], [28]. The computation at the control nodes occurs in parallel, and in each iteration a control node performs only a single linear computation given by (8). Given that each Boolean computation incurs a constant complexity, i.e., $\mathcal{O}(1)$, the total time complexity of a control node is $\mathcal{O}(1)$. On the other hand, variable nodes in our factor graph model simply send out the value of their current state. Similarly, a variable node incurs a constant time complexity of $\mathcal{O}(1)$. Therefore, the time complexity of our proposed model is $\mathcal{O}(1)$ per iteration for each pair of control node and variable node. For a constant number of iterations, the complexity of our model is proportional to the number of nodes in the graph, i.e., of order $\mathcal{O}(n)$. In large biological networks where few highly connected nodes regulate large poorly connected nodes, the complexity of our model is linear in the number of nodes.

2) MODELS COMPARISON

In the literature, noise propagation in biological systems has been studied theoretically and experimentally. For example, in [17], the authors explored noise propagation in genetic cascades using single-cell measurements. Hooshangi and Weiss [43] also computationally and theoretically analyzed noise propagation in linear cascades and reported that the existence of feedback loops in biological systems may provide robustness to extrinsic noise. They further showed that it is impossible to achieve lower-than-intrinsic noise control for n-ring gene regulatory networks with an odd number of nodes and negative regulations. In their work [19], Zhang *et al.* performed bifurcation analyses of ordinary differential equations for the Myc/E2F/miR-17-92 network [44], in order to study the role of feedback loop in sensitivity and noise amplification on the dynamic properties of the system.

These previous works mostly depend on the knowledge of biochemical and kinetic reactions which may be difficult to obtain for large genetic networks. Moreover, they focus mainly on investigating noise propagation in biological systems by considering how feedback loops impact noise in the system. These investigations have provided insights into how to tune or control noise; however, a study on how to quantify noise in large arbitrary genetic networks without the explicit determination of biochemical equations is lacking, thereby obscuring a direct quantitative comparison with our proposed model.

3) APPLICATION TO HUB-LIKE GRNs

The general framework of factor graphs is a powerful tool that has been employed in many fields of research [25], [26], [28], [32]. For example, in coding theory, factor graphs have been used to model networks of 100, 000 nodes and more, due to the sparse distribution of nodes in these networks. Similarly, a key experimental observation is that large biological graphs have sparsely distributed and possibly long edges [41], [42], i.e., a few highly connected nodes known as hubs regulate large, poorly connected nodes. In regulatory hub-like networks or large biological networks, we can adapt (14) by including higher-order terms in the Taylor series expansion of (12) as well as increasing the maximum allowable k_a and k_r values to obtain a more refined necessary condition in terms of $\beta_{u,v}$ for vanishing errors or noise.

IV. CONCLUSION

Here, we have explored a graphical model representation of biological networks and applied message passing to evolve errors or disturbances in such networks. Subsequently, we provided a density evolution analysis to study the behavior of biological systems in the presence of noise. The derived closed-form expressions enable an analytic approach to quantify and characterize the evolution of errors due to randomness in gene expressions. Our analysis resulted in a necessary condition on the network parameters for network resilience against perturbations. In other words, given the activation-inhibition Boolean functions implemented to model the cooperative effect biological rules, our derived necessary condition can allow us to constrain network parameters such as connectivity in designing reliable and noise-tolerant artificial gene circuits. Our work and approach provide a step towards understanding error propagation in complex genetic graphs with the aim of inciting research on noise control and intervention strategies for noisy biological systems. Finally, a possible future path would be to consider error propagation in factor graph models with unique computational rules on each control node.

APPENDIX

After a random perturbation or disturbance that introduces an error with probability ϵ has occurred, we can describe the probability that an error exists on node x at the beginning of the l-th iteration in the form of (9). Under the assumption that the error events on x_a and x_r are independent and identically distributed (i.i.d), we have $\Pr(x_a, x_r) = \Pr(x_a) \cdot \Pr(x_r)$. To solve the first part on the right-hand side of (9a), i.e., $\Pr(\epsilon_x | x_a = 0, x_r = 0)$, we note that there is an error in x, if and only if (iff) both x_a and x_r have errors. Using the third Boolean function truth table in Table 2, only when both $x_a \to 1$ and $x_r \to 1$ does the output state change from the expected 0 state to 1. In this particular case, the input error on either x_a or x_r is not masked by the other. Hence, we represent the first part of (9a) as follows:

$$\Pr(\epsilon_x | x_a = 0, x_r = 0) = \Pr(\epsilon_{x_a} | x_a = 0) \cdot \Pr(\epsilon_{x_r} | x_r = 0). \tag{17}$$

Now consider the network configuration shown in Fig. 5 with two activating edges and two inhibiting edges. We assume that an error is introduced with a positive probability $\epsilon \ll 1$ by which the state of nodes can randomly be changed. Using the Boolean truth table for activation in Table 2, there will be an error in x_a if at least one of the edges x_1 and x_2 has an error, giving $\Pr(\epsilon_{x_a} | x_a = 0) = 1 - (1 - \epsilon)^2$. In general, for a node with k_a activating edges

$$\Pr(\epsilon_{x_a} | x_a = 0) = 1 - (1 - \epsilon)^{k_a}. \tag{18}$$

Similarly, using the Boolean truth table for repression, we compute the probability of error in x_r by

$$\Pr(\epsilon_{x_r} | x_r = 0) = \frac{1}{2^{k_r}} \left[1 - (1 - \epsilon)^{k_r} \right], \tag{19}$$

where k_r denotes the number of repression edges on a node. Subsequently, we compute the probabilities $\Pr(x_a = 0) = \frac{1}{2^{k_a}}$ and $\Pr(x_r = 0) = \frac{2^{k_r} - 1}{2^{k_r}}$. Hence, we write the expression for (9a) as

$$\begin{aligned} &\Pr(\epsilon_x | x_a = 0, x_r = 0) \cdot \Pr(x_a = 0, x_r = 0) \\ &= \frac{1}{2^{k_a}} \cdot \frac{2^{k_r} - 1}{2^{k_r}} \cdot \frac{1}{2^{k_r}} \left[1 - (1 - \epsilon)^{k_a} \right] \\ &\quad \times \left[1 - (1 - \epsilon)^{k_r} \right]. \end{aligned} \tag{20}$$

We next compute the probability expression for (9b). From the third Boolean truth table in Table 2, we observe that there will be an error in x iff $x_r \to 1$ and no error on x_a. The probability of error in x_r, i.e., $\Pr(\epsilon_{x_r} | x_r = 0)$, is given by (19). Using the activation and repression truth tables in Table 2 and considering the general case of a node with k_a and k_r input edges, we compute the following probability of error:

$$\Pr(\epsilon_{x_a} | x_a = 1) = \frac{1}{2^{k_a}} \left[1 - (1 - \epsilon)^{k_a} \right]. \tag{21}$$

Therefore, $\Pr(\overline{\epsilon}_{x_a} | x_a = 1) = 1 - \Pr(\epsilon_{x_a} | x_a = 1)$, where $\overline{\epsilon}_{(\cdot)}$ denotes the event where there is no error. In addition, we compute the probability $\Pr(x_a = 1) = \frac{2^{k_a} - 1}{2^{k_a}}$ and then write the expression for (9b) as

$$\begin{aligned} &\Pr(\epsilon_x | x_a = 1, x_r = 0) \cdot \Pr(x_a = 1, x_r = 0) \\ &= \Pr(\overline{\epsilon}_{x_a} | x_a = 1) \cdot \Pr(\epsilon_{x_r} | x_r = 0) \cdot \Pr(x_a = 1, x_r = 0) \\ &= \frac{2^{k_a} - 1}{2^{k_a}} \cdot \frac{2^{k_r} - 1}{2^{k_r}} \left\{ 1 - \frac{1}{2^{k_a}} \left[1 - (1 - \epsilon)^{k_a} \right] \right\} \\ &\quad \times \frac{1}{2^{k_r}} \left[1 - (1 - \epsilon)^{k_r} \right]. \end{aligned} \tag{22}$$

Likewise, we can compute the probability of error in (9c) by

$$\begin{aligned} &\Pr(\epsilon_x | x_a = 0, x_r = 1) \cdot \Pr(x_a = 0, x_r = 1) \\ &= \Pr(\epsilon_{x_a} | x_a = 0) \cdot \Pr(\overline{\epsilon}_{x_r} | x_r = 1) \cdot \Pr(x_a = 0, x_r = 1) \\ &= \frac{1}{2^{k_a}} \cdot \frac{1}{2^{k_r}} \left[1 - (1 - \epsilon)^{k_a} \right] (1 - \epsilon)^{k_r}, \end{aligned} \tag{23}$$

where an error results in x iff $x_a \to 1$ and no error on x_r.

Finally, to solve (9d), we note that there will be an error in x if at least one of the inputs in the third Boolean truth table of Table 2 has an error, i.e., the output $x_a \wedge x_r$ changes state from 1 to 0. Hence, we express the first part of the right-hand side of (9d) as:

$$\begin{aligned} &\Pr(\epsilon_x | x_a = 1, x_r = 1) \\ &\quad = \Pr(\overline{\epsilon}_{x_a} | x_a = 1) \cdot \Pr(\epsilon_{x_r} | x_r = 1) \\ &\qquad + \Pr(\epsilon_{x_a} | x_a = 1) \cdot \Pr(\overline{\epsilon}_{x_r} | x_r = 1) \\ &\qquad + \Pr(\epsilon_{x_a} | x_a = 1) \cdot \Pr(\epsilon_{x_r} | x_r = 1). \end{aligned} \tag{24}$$

Furthermore, we express the error probability $\Pr(\epsilon_{x_r} | x_r = 1) = 1 - (1 - \epsilon)^{k_r}$. Therefore, (24) yields

$$\begin{aligned} &\Pr(\epsilon_x | x_a = 1, x_r = 1) \\ &\quad = \left\{ 1 - \frac{1}{2^{k_a}} \left[1 - (1 - \epsilon)^{k_a} \right] \right\} \left[1 - (1 - \epsilon)^{k_r} \right] \\ &\qquad + \frac{1}{2^{k_a}} \left[1 - (1 - \epsilon)^{k_a} \right] (1 - \epsilon)^{k_r} \\ &\qquad + \frac{1}{2^{k_a}} \left[1 - (1 - \epsilon)^{k_a} \right] \left[1 - (1 - \epsilon)^{k_r} \right]. \end{aligned} \tag{25}$$

Then, we can express the probability of error in (9d) as

$$\Pr(\epsilon_x|x_a=1,x_r=1)\cdot\Pr(x_a=1,x_r=1) = \frac{2^{k_a}-1}{2^{k_a}}\cdot\frac{1}{2^{k_r}}\cdot\Pr(\epsilon_x|x_a=1,x_r=1). \quad (26)$$

Inserting (20), (22), (23), and (26) into (9) and after simplification, we obtain the total probability of error on node x as

$$\begin{aligned}\epsilon_l = \frac{1}{2^{2k_i}}\Big\{&2^{k_i}\left(2^{k_a}-1\right)-\Big[\left(2^{k_a+1}-1\right)(1-\epsilon)^{k_a}\\&-3\cdot 2^{k_a}+4^{k_a}+1\Big](2-2\epsilon)^{k_r}\\&-\left(2^{k_r}-1\right)\left[1-(1-\epsilon)^{k_a}-2^{k_a}+4^{k_a}\right]\\&\times\left[(1-\epsilon)^{k_r}-1\right]\Big\}, \end{aligned} \quad (27)$$

where $k_i = k_a + k_r$.

Conflict of Interest: none declared.

REFERENCES

[1] L. S. Tsimring, "Noise in biology," *Rep. Prog. Phys.*, vol. 77, no. 2, Jan. 2014, Art. no. 026601.

[2] P. S. Swain, M. B. Elowitz, and E. D. Siggia, "Intrinsic and extrinsic contributions to stochasticity in gene expression," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 20, pp. 12795–12800, Oct. 2002.

[3] Z. Wang and J. Zhang, "Impact of gene expression noise on organismal fitness and the efficacy of natural selection," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 16, pp. E67–E76, Apr. 2011.

[4] D. Noble, "The role of stochasticity in biological communication processes," *Prog. Biophys. Mol. Biol.*, vol. 162, pp. 122–128, Oct. 2020.

[5] T. M. Venancio, S. Balaji, S. Geetha, and L. Aravind, "Robustness and evolvability in natural chemical resistance: Identification of novel systems properties, biochemical mechanisms and regulatory interactions," *Mol. BioSyst.*, vol. 6, no. 8, pp. 1475–1491, Jun. 2010.

[6] J. M. Raser and E. K. O'Shea, "Control of stochasticity in eukaryotic gene expression," *Science*, vol. 304, no. 5678, pp. 1811–1814, Jun. 2004.

[7] W. J. Blake, M. Kærn, C. R. Cantor, and J. J. Collins, "Noise in eukaryotic gene expression," *Nature*, vol. 422, no. 6932, pp. 633–637, Apr. 2003.

[8] V. Kohar and M. Lu, "Role of noise and parametric variation in the dynamics of gene regulatory circuits," *npj Syst. Biol. Appl.*, vol. 4, no. 1, pp. 1–11, Nov. 2018.

[9] B. Munsky, G. Neuert, and A. van Oudenaarden, "Using gene expression noise to understand gene regulation," *Science*, vol. 336, no. 6078, pp. 183–187, Apr. 2012.

[10] M. Kittisopikul and G. M. Suel, "Biological role of noise encoded in a genetic network motif," *Proc. Nat. Acad. Sci. USA*, vol. 107, no. 30, pp. 13300–13305, Jul. 2010.

[11] D. Frigola, L. Casanellas, J. M. Sancho, and M. Ibañes, "Asymmetric stochastic switching driven by intrinsic molecular noise," *PLoS ONE*, vol. 7, no. 2, Feb. 2012, Art. no. e31407.

[12] T. Biancalani, L. Dyson, and A. J. McKane, "Noise-induced bistable states and their mean switching time in foraging colonies," *Phys. Rev. Lett.*, vol. 112, no. 3, Jan. 2014, Art. no. 038101.

[13] M. Assaf, E. Roberts, Z. Luthey-Schulten, and N. Goldenfeld, "Extrinsic noise driven phenotype switching in a self-regulating gene," *Phys. Rev. Lett.*, vol. 111, no. 5, Jul. 2013, Art. no. 058102.

[14] G. Tkačik, A. M. Walczak, and W. Bialek, "Optimizing information flow in small genetic networks. III. A self-interacting gene," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 85, no. 4, Apr. 2012, Art. no. 041903.

[15] A. M. Walczak, G. Tkačik, and W. Bialek, "Optimizing information flow in small genetic networks. II. Feed-forward interactions," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 81, no. 4, Apr. 2010, Art. no. 041905.

[16] G. Tkacik, C. G. Callan, and W. Bialek, "Information flow and optimization in transcriptional regulation," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 34, pp. 12265–12270, Aug. 2008.

[17] J. M. Pedraza and A. van Oudenaarden, "Noise propagation in gene networks," *Science*, vol. 307, no. 5717, pp. 1965–1969, Mar. 2005.

[18] L. Arola-Fernández, G. Mosquera-Doñate, B. Steinegger, and A. Arenas, "Uncertainty propagation in complex networks: From noisy links to critical properties," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 30, no. 2, Feb. 2020, Art. no. 023129.

[19] H. Zhang, Y. Chen, and Y. Chen, "Noise propagation in gene regulation networks involving interlinked positive and negative feedback loops," *PLoS ONE*, vol. 7, no. 12, Dec. 2012, Art. no. e51840.

[20] G. Hornung and N. Barkai, "Noise propagation and signaling sensitivity in biological networks: A role for positive feedback," *PLoS Comput. Biol.*, vol. 4, no. 1, p. e8, Jan. 2008.

[21] M. L. Simpson, C. D. Cox, and G. S. Sayler, "Frequency domain analysis of noise in autoregulated gene circuits," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 8, pp. 4551–4556, Apr. 2003.

[22] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems Biology in Practice: Concepts, Implementation and Application*. New Jersey, NJ, USA: Wiley, 2008, Ch. 5, pp. 137–145.

[23] S. Martin, Z. Zhang, A. Martino, and J.-L. Faulon, "Boolean dynamics of genetic regulatory networks inferred from microarray time series data," *Bioinformatics*, vol. 23, no. 7, pp. 866–874, 2007.

[24] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*. New York, NY, USA: Oxford Univ. Press, 1993, Ch. 12, pp. 441–480.

[25] S. Kotiang and A. Eslami, "Boolean factor graph model for biological systems: The yeast cell-cycle network," *BMC Bioinf.*, vol. 22, no. 1, pp. 1–27, Dec. 2021.

[26] A. Behfarnia and A. Eslami, "Error correction coding meets cyber-physical systems: Message-passing analysis of self-healing interdependent networks," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2753–2768, Jul. 2017.

[27] I. Gat-Viks, A. Tanay, D. Raijman, and R. Shamir, "A probabilistic methodology for integrating knowledge and experiments on biological networks," *J. Comput. Biol.*, vol. 13, no. 2, pp. 165–181, Mar. 2006.

[28] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[29] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," *Nature*, vol. 9, pp. 770–780, Oct. 2008.

[30] S. Kotiang and A. Eslami, "A probabilistic graphical model for system-wide analysis of gene regulatory networks," *Bioinformatics*, vol. 36, no. 10, pp. 3192–3199, May 2020.

[31] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle," *Bioinformatics*, vol. 22, no. 14, pp. e124–e131, Jul. 2006.

[32] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[33] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2005, pp. 22–90.

[34] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: MKP, 2014, Ch. 3, pp. 116–126.

[35] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.

[36] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 891–907, May 2001.

[37] F. Liu, S. W. Zhang, W. F. Guo, Z. G. Wei, and L. Chen, "Inference of gene regulatory network based on local Bayesian networks," *PLoS Comput. Biol.*, vol. 12, no. 8, Aug. 2016, Art. no. e1005024.

[38] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm," *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, Oct. 2006.

[39] K. Basso, A. A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera, and A. Califano, "Reverse engineering of regulatory networks in human B cells," *Nature Genet.*, vol. 37, no. 4, pp. 382–390, Apr. 2005.

[40] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, "The yeast cell-cycle network is robustly designed," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 14, pp. 4781–4786, Apr. 2004.

[41] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp. 102–105, Jul. 2003.

[42] J. Tegner, M. K. S. Yeung, J. Hasty, and J. J. Collins, "Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 10, pp. 5944–5949, May 2003.
[43] S. Hooshangi and R. Weiss, "The effect of negative feedback on noise propagation in transcriptional gene networks," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 16, no. 2, Jun. 2006, Art. no. 026108.
[44] Y. Li, Y. Li, H. Zhang, and Y. Chen, "MicroRNA-mediated positive feedback loop and optimized bistable switch in a cancer network involving miR-17-92," *PLoS ONE*, vol. 6, no. 10, Oct. 2011, Art. no. e26302.

STEPHEN KOTIANG (Member, IEEE) received the B.Eng. degree in electrical and communications engineering from Moi University, Eldoret, Kenya, in 2010, and the M.Eng. degree in electronic engineering from Chonbuk National University, Jeonju, Republic of Korea, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS, USA. Since 2011, he has been a Lecturer with the Electrical Engineering Department, Moi University. His research interests include probabilistic graphical models, applications of machine learning and graphical modeling in systems biology, and bioinformatics.

ALI ESLAMI (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 2013. He was a Postdoctoral Research Fellow with Texas A&M University, College Station, TX, USA, from March 2013 to April 2015. From August 2014 to June 2015, he was a Visiting Research Scholar as part of the information initiative at Duke (iiD). He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Wichita State University, Wichita, KS, USA. His current research interests include the application of probabilistic methods and machine learning to study complex networks, such as gene regulatory networks and networked cyber-physical systems. He is a member of the IEEE Communications and Computer Societies. From 2016 to 2017, he was a recipient of Wichita State University's Young Faculty Risk Taker Award. He has served as the Session Chair for several IEEE conferences and workshops, and as a reviewer for numerous IEEE journals. He has also served on several NSF review panels.