Delayed and Hidden Variables Interactions in Gene Regulatory Networks

Mina Moradi Kordmahalleh,
Mohammad Gorji Sefidmazgi,
Abdollah Homaifar*, Ali Karimoddini
Dep. Electrical & Computer Eng.
North Carolina A&T State University
Greensboro, NC, USA

Anthony Guiseppi-Elie
Dep. of Electrical & Computer Eng.
Clemson University
Clemson, SC, USA

Joseph L. Graves Jr.
Joint School
Nanoscience & Nanoengineering
North Carolina A&T State University
UNC Greensboro
Greensboro, NC, USA

***Abstract*— Reverse Engineering of Gene Regulatory Networks (GRN), i.e. finding appropriate mathematical models to understand complex cellular systems, can be used in disease diagnosis, treatment, and drug design. There are fundamental gaps in the construction of GRN with regard to modeling of hidden/delayed interactions. Addressing these deficiencies is critical to understanding complex intracellular processes and enabling full use of the vast and ever-growing amount of available genomic data. Current modeling strategies either ignore or oversimplify time delays resulted from transcription and translation processes during gene expression. In addition, many research works do not account hidden variables such as transcription factors, repressors, small metabolites, DNA, microRNA species that regulate themselves and other genes but are not readily detectable on microarray experiments. To capture the effect of these parameters, in this paper, we utilize our developed Partially Connected Artificial Neural Networks with Evolvable Topology (PANNET) to find a more comprehensive model of GRN by considering the effects of unknown hidden variables and different time delays. This method is innovative, since the structure of the network has memory and internal states, which can model the unknown hidden variables and time delays. We furthermore use a new evolutionary optimization based on variable-length Genetic Algorithm (GA) to find a sparse structure of PANNET to predict the gene expression levels accurately. Finally we demonstrate the capability of PANNET in constructing GRN, including the effect of different delays and unknown hidden variables through modeling of E. coli SOS inducible DNA repair system.**

Keywords—gene regulatory network; artificial neural network genetic algorithm; evolvable topology; time series

I. INTRODUCTION

Accurate mathematical models that describe the interactions within and between groups of genetic and regulatory cellular components (i.e. Gene Regulatory Network) are essential for fully understanding the complex biological systems involved in cellular differentiation, cancer development, aging, disease etiology and response to therapy. The ability to accurately model GRN activities enables us to make full use of the burgeoning accumulation of genome-wide expression data and bridge the gap between data and knowledge [1], [2].

Understanding complex cellular systems requires studying how their different components work together. Since cellular activity measurements are available at the genomic scale, mathematical modeling of complex biological networks (known as Reverse Engineering) reveals an overall picture of biological processes that take place in a cell. DNA microarray technology enables us to simultaneously measure the expression levels of thousands of genes inside GRN as they respond to specific environmental conditions [3]. Defining the internal structure and functions of a GRN requires gene expression time series which can be provided by collecting cell or tissue samples over several time instants. Constructing a GRN model using temporal gene expression levels is crucial in targeted in-silico experiments to investigate and predict the behavior of the system under different conditions. Such studies have led to important improvements in medical diagnosis, disease treatment, and drug design [4].

Modeling the temporal dimension of GRN is a challenge, since different delays exist due to the time required for a regulatory gene to express its protein product and for the transcription of the target genes to be affected by these regulatory proteins [5]. Especially, each one of these interactions may have different time lags [6]. In addition, the GRN model has to effectively account for the presence and activities of regulatory intermediates, such as microRNAs, proteins, and metabolites. These "hidden variables" are not always represented in microarray data [7], [8], but can introduce different delays into the regulatory response time. Hence, the consideration of different time delays in the presence of unknown hidden variables in GRN is a critical issue.

Most of the current methods of GRN modeling ignore or oversimplify the effect of time delays and hidden variables, and are not adequate for finding reliable models to understand dynamical behaviors of complex biological systems. For example, some methods construct the GRN without considering any time delay [9], [10], [11]. Some other methods incorporate a fixed time delay that is found based on a correlation matrix [12], stochastic simulation algorithms (SSA) [13], [14], or dynamic Bayesian networks [6]. [15] and [16] employ a model based on delayed differential equations, but in these works, the time delay parameters are set arbitrarily rather than being derived from experimental observations.

Canonical Recurrent Neural Networks (RNN) are networks with delayed feedbacks and used for modeling the temporal behaviors of gene expression data. However, studies based on

This material is based in part upon work supported by the National Science Foundation (NSF) under Cooperative Agreement No. DBI-0939454. Any opinions, findings and conclusions are those of the authors and do not necessarily reflect the views of the National Science Foundation.*Email of Corresponding author: homaifar@ncat.edu

978-1-4799-7502-0/14 $31.00 © 2014 IEEE
DOI 10.1109/BIBE.2014.40

RNN either consider fixed time-delays or do not consider the effect of hidden variables which operate within GRN [17], [18], [19]. El Bakry et al. [20] presented an approach, based on pairwise correlations, to infer GRN variable time delays without considering the effect of hidden variables. Estimation of known hidden variables (some identified proteins and transcription factors) has been investigated by state space modeling to infer hidden state variables from observations [7], [21]. The Unscented Kalman Filter is employed for estimation of both parameters and identified hidden variables when the nonlinear state space model is known [7]. However, because of experimental limitations, there are still some unknown hidden variables that influence GRN, which are not addressed in literature. Thus, it is not easy to find state space GRN models to infer unknown hidden variables. In general, the current methods still did not take into consideration unknown hidden variables and different delays between various parameters, which are inherent properties of GRN. To address this problem, we aim at incorporating different delays and unknown hidden variables to provide a comprehensive and more accurate interpretation of internal interactions in GRN.

In this paper, we develop a method that incorporates the effects of different time delays and unknown hidden variables into mathematical models of GRN. The proposed approach is based on Partially Connected Artificial Neural Networks with Evolvable Topology (PANNET), which have been developed in our recent work for modeling and forecasting behavior of chaotic systems [22]. In the case of GRN, PANNET constructs a nonlinear mathematical model between gene measurements called as observation nodes, and additional un-measured variables called as hidden nodes. Using the evolutionary cycle of a proposed variable-length Genetic Algorithm (GA) with novel crossover and mutation operators, the topology of the network is evolved. Through the evolution, a sparse or partially connected topology of the network is generated to model the underlying behavior of the system while hidden nodes play the role of memory and internal states in the network.

This rest of this paper is organized as follows. Section II describes proposed PANNET method and evolutionary procedure for generating an efficient topology. In section III, the simulation results are presented. Section IV concludes the paper.

II. Method Description

Artificial Neural Networks (ANNs) are parallel processors that can approximate nonlinear and complicated functions. They are particularly useful when analytical solutions are not possible, or very difficult, using conventional methods [23], [24]. The common structure of ANNs consist of a set of input/output nodes, and internal processing units i.e. neurons distributed in different layers, where the strength of connections between nodes and neurons are determined through learning procedures using available observations. However common structure of ANNs are not suitable in GRN modeling, since a GRN cannot be solely characterized by external inputs (if exists) and gene observations, if unknown hidden variables, internal states, and time delay play an important role in the dynamics of genetic networks [7], [20].

The focus of this paper is on the application of the proposed PANNET for modeling delayed/hidden interactions in GRN. In this context, a novel Genetic Algorithm as a bio-inspired computational algorithm is used to adapt the parameters and topology of the network. This methodology leads to a partial connected i.e. sparse configuration between external signal (if exist), delayed gene expressions and a set of unknown hidden variables. Hidden nodes can play the role of memory and internal states depending on the complexity of the system. This capability is useful in GRN modeling due to the fact that GRN are sparse, i.e. genes are regulated by limited number of other genes [25]; and also PANNET can be trained with fewer number of temporal sample points [26] (which is common in microarray). To this aim, the basic concept and modeling approach of the PANNET for constructing the GRN and forecasting the gene profiles are introduced.

A. The PANNET Approach to GRN

In this section, we illustrate the applicability of PANNET for GRN modeling to capture the relationship among external input, gene observation nodes and hidden nodes to provide accurate prediction of biological systems. From the modeling point of view, the time delays and unknown hidden variables that are taking place in the middle of observed gene regulation are to be considered. PANNET consists of an arbitrary number of neurons, which are partially connected to *external input nodes*, *hidden nodes*, and *observation nodes* (Fig. 1). Where external inputs represent the externally added chemicals, nutrients, or other exogenous inputs [27], observation nodes represent current values of gene expression level; and hidden nodes play the role of memory or internal states of the system that we don't have experimental measurements for them. In this case, the only available information is that the number of hidden variables are much smaller than the number of transcribed genes and most of the genes are regulated by a small number of transcription factors [8]. To determine the topology of PANNET, we used the evolutionary cycle of a proposed variable-length GA with novel crossover and mutation operators. The topology of PANNET is characterized by the number of neurons, the number and origin of the input/output nodes for each neuron, and connection weights, which are supposed to evolve based on the defined fitness of GA. The fitness is an index function that determines the accuracy of the extracted PANNET model. Here, this fitness is considered as the Mean Squared Error between actual values of observation nodes and their estimated values by PANNET. In the last generation of this evolutionary process, a nonlinear state space mapping is found between external input, observation nodes and hidden nodes.

Fig. 1 illustrates the structure of a typical PANNET, which consists of one external input node, x_1, two hidden nodes, x_2 and x_3, and three gene observation nodes, x_4, x_5 and x_6. Assume that the nodes are partially connected in the network via two neurons. For example, the first neuron (*Neuron1*) is connected to three inputs x_1, x_2, x_4 , at time t with corresponding weights w_1^1, w_2^1, w_4^1 , and three outputs x_2, x_4, x_5, at time $t+1$ with the corresponding weights v_2^1, v_4^1, v_5^1. In order to find how the gene observation nodes and hidden nodes are updated from

time t to $t+1$, the following two equations are used to obtain the value of the $x_i(t+1)$:

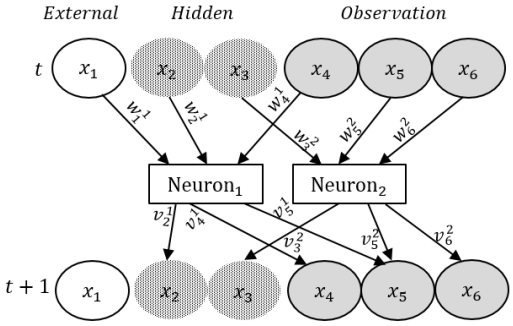


Fig. 1. A typical PANNET structure includes one external input, two hidden nodes, three observation nodes and two neurons

$$x_i(t+1) = \sum_k v_i^k . f(\sum_j w_j^k . x_j(t)) \quad (1)$$

$j \in \{$ Incoming nodes to Neuron k$\}$, $i \in \{$Hidden, observation$\}$

$$f(t)=1/(1+e^{-t}) \quad (2)$$

where i is the index of updated hidden nodes and observation nodes, w_j^k is the weight of connection from node $x_j(t)$ to k^{th} neuron and v_i^k is the weight of connection from k^{th} neuron to i^{th} node at time $t+1$. In case of multiple incomings connections from different neurons to the node x_i, the value of node $x_i(t+1)$ has to be revised by adding the results coming from different neurons. With this set up, the problem now is to find number of neurons, connection weights, the number and origin of the inputs and outputs for each neuron. In other words, the structure of network including connections between the nodes is not known. To find the unknown parameters, we introduce a novel GA through in which a set of candidate network topologies (individuals or candidate solutions) will be evolved so that the network structure will be eventually adapted to a suitable topology with minimum possible fitness.

B. Evolutionary Algorithm

Candidate solutions in genetic algorithm are a concatenation of a random number of neurons along with their description which represent the network topologies. In each individual, number of neurons, number of input nodes, number of output nodes, origin of the input nodes, origin of the output nodes, connection weight of the input nodes, and the connection weight of the output nodes for each neuron has to be selected. These parameters are indicated as N_N, N_{In}, N_{out}, In, Out, In_w, Out_v respectively. Before evolutionary process, initial number of neuron and hidden nodes are decided by the design of the experiment. Fig. 2 illustrates the first and last neurons of a candidate network consisting of K neurons.

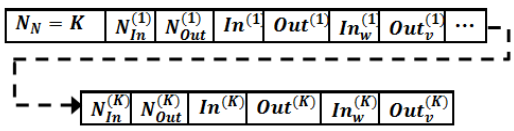


Fig. 2. Candidate structures of PANNET with K neurons

The performance of the candidate network topology is evaluated by a fitness function, which is defined as the mean sum square error between all P gene observation nodes x_i and their corresponding estimation $\hat{x}_i$, over the whole set of training data with T samples, and L different set of experiments as follows.

$$Fitness = \frac{1}{T.P.L} \sum_{l=1}^{L} \sum_{t=1}^{T} \sum_{i} (x_i(t) - \hat{x}_i(t))^2 \quad (3)$$

Different candidate network topologies with different number of neurons and different inputs/outputs for each neuron will lead to GA with different length of individuals. Because of that, traditional crossover and mutation operators are not appropriate. Thus, we have developed specific genetic operators for variable-length GA. These operators evolve the topology of the networks and generate new offspring from candidate individuals with different lengths. Crossover with probability of P_c swaps one arbitrary selected neuron in one individual with all its components including its corresponding input/output nodes and connection weights with the one in the second individual with its corresponding components. In other words, this is a type of two-point crossover where crossover sites are at the beginning and the end-points of a neuron and may change the length of the individuals.

The proposed mutation with probability of P_m mutates the number of neurons, i.e. N_N, or one of the their components including. N_{In}, N_{Out}, In, Out, In_w and Out_v. Mutation on the number of neurons is equivalent to searching new region and causes bigger changes in the solution space; while mutation on one of the parameters of neuron is similar to searching in the neighborhood of a local solution. Therefore, the developed mutation operator insured searching of the local space and escaping from possible local minimums.

Mutation on the N_N will delete/add a neuron in the network. If this new number of N_N is higher than the previous one, a new neuron with a set of its random input/output connections have to be added to the network. Otherwise, one of the existing neurons in the individual along with all its connections have to be removed. Mutation at N_{In}, N_{Out}, In, Out, and In_w and Out_v generally leads to adding/deleting an input/output node and its corresponding connection weight in the neuron, or changing in the wiring or weight of a connection.

If the mutation site is on N_{In}, and the mutated N_{In} is more than the old one, a random new input and its corresponding random weight have to be added to the neuron in the network. Otherwise, an input and its corresponding weight have to be removed from the neuron. Procedure for mutation on N_{Out} is similar to mutation of N_{In}.

If the mutation site is on one the nodes in In i.e. input vector of a neuron, its corresponding connection is rewired to another node. Procedure for mutation on Out is similar to mutation of In. This is a way to change the connections between nodes and neurons to improve the capability of the network based on its partially connected structure.

If the mutation site is on the weights vector [In_w, Out_v], the selected weight $W \in \{w, v\}$ is mutated according to (4), where α is a random value in the range of [0, 1].

$$W = W(1 \pm \alpha) \quad (4)$$

Since, the number of mutation sites in the defined individual is equal to the length of the individual, then the rate of the mutation on N_N which is equal to 1/(length of individual) is smaller than the rate of mutation on other components of the individual. This is aligned with [28], which states that most of the improvements in evolutionary progress are related to searching the neighborhood of the local region.

By applying the crossover and mutation operators, individuals that represent the configurations of the networks are evolved. It is worth mentioning that based on the capability of PANNET, a near optimal partially connected network will be generated. Moreover, this topology includes non-uniform time delays or internal states that model the underlying dynamical behavior of the biological system with efficient accuracy of prediction.

III. Experimental Results on the Escherichia coli SOS DNA Repair System

In this section, we illustrate the capability of PANNET in constructing GRN, since the effect of different delays and unknown hidden variables are taking into consideration in modeling. To this aim, we selected expression profile of the E. coli SOS inducible DNA repair system [29].

The E. coli SOS system is a well-characterized bacterial gene network, consisting of about thirty transcriptionally-regulated genes. Under normal conditions, a constitutively-expressed master repressor protein *LexA* represses the expression of the genes responsible for DNA repair. However, when, the SOS sensor protein *RecA* encounters and binds damaged (single-strand) DNA, it becomes activated and triggers auto cleavage of *LexA*. The subsequent drop in *LexA* level results in derepression and subsequent upregulation of an array of SOS genes. Once DNA damage has been repaired, the level of activated *RecA* drops, allowing *LexA* to reaccumulate in the cell and repress the SOS genes. At this point, the cells return to their original pre-induced state (Fig. 3) [27].

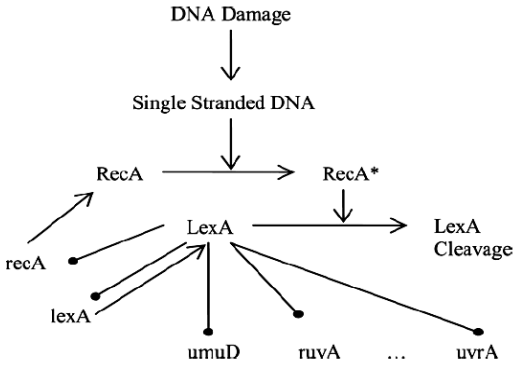


Fig. 3. The bacterial E. coli SOS DNA Repair network [27].
Inhibitions are represented by–•, while activations are represented by →

For this study, we used temporal expression data on eight SOS genes: *uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA*, and *polB*. Gene expression was quantified using plasmid-borne gfp (Green Fluorescent Protein) chimeric reporter genes and was measured at 6 minute intervals over 5 hour (50 time points) following induction of the system by ultraviolet irradiation. The data set comprises two sets of duplicate experiments, in which the SOS response was induced by two different UV doses (5 Jm−2 in experiments 1 and 2, and 20 Jm-2 in experiments 3 and 4) [30]. Fig. 4 shows the corresponding temporal gene expression levels of each of the eight SOS genes related to experiment 1.

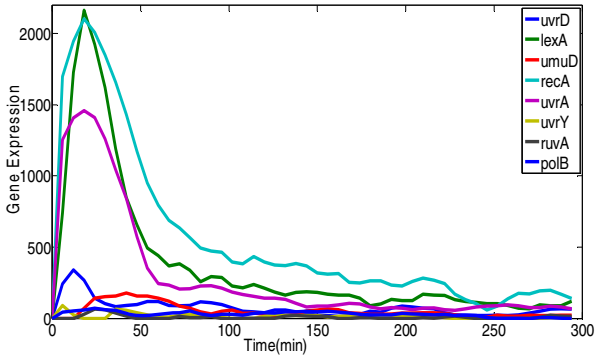


Fig. 4. Measured time series of 8 genes in the *E. coli* SOS DNA repair system (Experiment1 with UV doses equal to 5 Jm^{-2}) [30].

For reconstructing the network with PANNET, we have used gene expression measurements of experiment 1 and 2 (5 Jm−2 UV induction). The first 35 time points of each experiment was considered for training the model of GRN and the remaining 15 time points are considered for testing the results. In this way networks are tested by the data with different time point comparing to the ones used as training. This can lead us to more accurate and realistic evaluation from our networks [31].

All eight time series in each experiment were normalized in the range [0, 1] with respect to a unique maximum-minimum. In the results, we have shown regulatory interactions for three target genes *lexA*, *recA,* and *uvrA*. For this purpose, we considered three different PANNET structures to model the behavior of three target genes *lexA, recA* and *uvrA* separately. Since there are not any exogenous inputs, input of each neuron can be selected from eight gene observation nodes, and a set of hidden nodes, while the output of each neuron can be selected from a target gene and hidden nodes. The initial numbers of hidden nodes are chosen by the designer which are 4, 4, 2 for networks of *lexA, recA, uvrA* respectively.

In order to evolve the networks configuration, parameters of GA are selected according to the table I. In this table, population size, number of generation, probability of crossover, probability of mutation, maximum number of inputs, and maximum number of outputs for the neurons are represented by N_p, N_g, P_c, P_m. *Max* N_{In}, and *Max* N_{Out}. Also, initial weights are randomly generated by a uniform distribution in the range of [-2, 2]. According to these parameters, plot of minimum fitness through the evolutionary

progress of GA for target genes *lexA, recA,* and *uvrA* are shown in Fig. 5.

TABLE I. PARAMETERS OF GA

N_p	P_c	Max. N_{In}	Elitism Rate
100	0.5	4	0.1
N_g	P_m	Max. N_{Out}	Tournament Size
2000	0.1	3	10

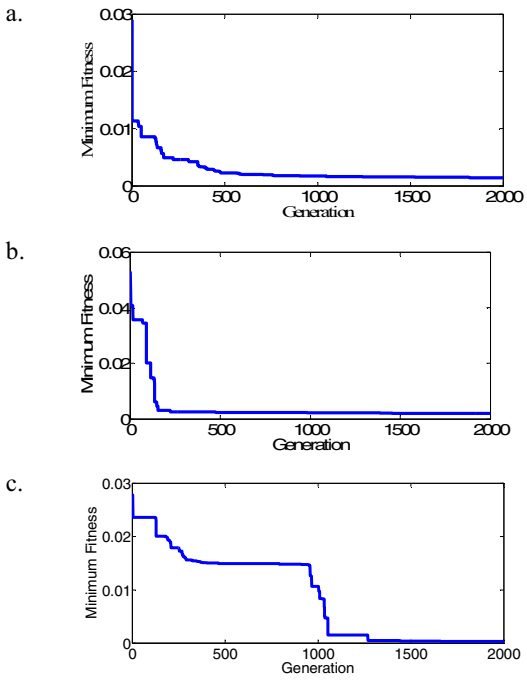


Fig. 5. Evolution progress of GA for target gene a. *lexA*, b. *recA,* c. *uvrA*

To see the efficiency of the developed PANNET in prediction of gene expression data, the predicted expression levels of target genes *lexA, recA,* and *uvrA* based on our proposed method are shown in Fig. 6. In each part of this figure, the experimentally determined gene expression profiles and the predicted expression profiles for both Exp. 1 and 2 are shown with solid blue line and broken red lines respectively. It can be seen that the resulting models have accurately interpreted the dynamics of the target genes *lexA*, *recA* and *uvrA* from the given experiments.

By selecting the parameters of table I, the generated networks for target gene *lexA, recA* and *uvrA* predict the test data of Exp.1 and 2 with the Normalized Root Mean Square Errors (NRMSE). See table II. NRMSE is calculated by (5) where x is the gene observed value, $\hat{x}$ is the estimated one, T is the total number of test samples, x_{min} and x_{max} are the minimum and maximum of the observed gene value respectively.

$$NRMSE = \frac{1}{(x_{\max} - x_{\min})}\sqrt{\frac{1}{T}\sum_{t=1}^{T}(x(t)-\hat{x}(t))^2} \qquad (5)$$

TABLE II. NRMSE OF THE PREDICTION

Target Gene	Exp. 1	Exp. 2
lexA	0.21	0.18
recA	0.21	0.22
uvrA	0.20	0.18

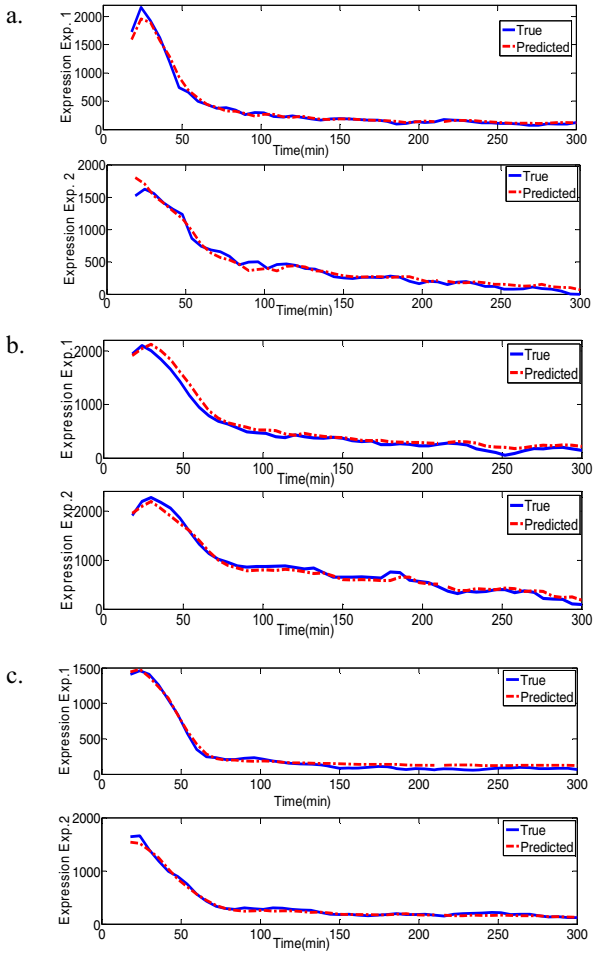


Fig. 6. Time series of measured and predicted expression of gene a. *lexA,* b. *recA*, c. *uvrA* (The first 35 time points are used for training, the remaining 15 time points are used for testing)

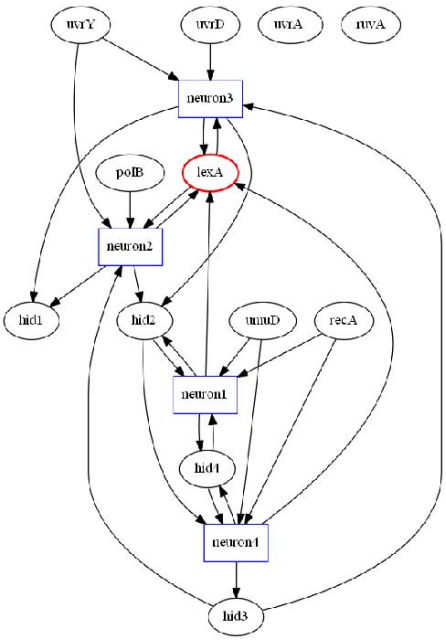
The constructed networks for these three genes are shown in Fig. 7 where hidden nodes are demonstrated with *hid*. The model of *lexA* from Fig. 7.a identifies the previously reported regulations of target gene *lexA* by *uvrD, umuD, lexA* and [31], [32], [33]. The model of *recA* from Fig. 7.b can identify the previously reported regulations of target gene *recA* by *umuD, lexA, recA* and *polB* [32], [33]. Also, model of *uvrA* in Fig. 7.c can identify the previously reported regulations of target gene *uvrA* by *lexA, uvrA* and *recA* [33], [35], [36]. These results demonstrated the strength of the proposed method in inferring real gene network topology.

As mentioned earlier, the hidden nodes act as memory or internal states in the model of the GRN. To make this idea more clear, consider the model of interactions for target genes

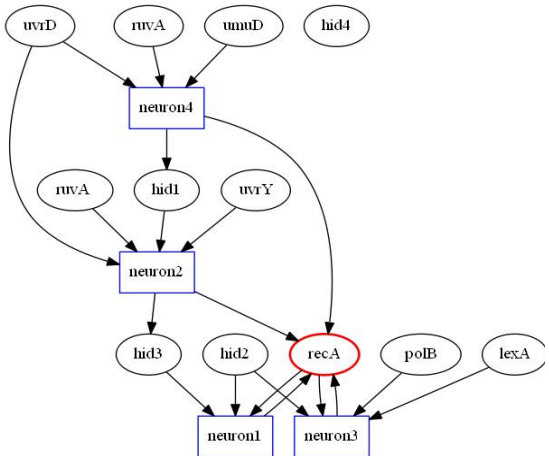
recA (Fig. 7.b), which seems more straightforward. It is obvious that *hid4* is not used at all. *hid1* is an inter-connection between *uvrD, uvrA*, and *umuD* at time *t* under a sigmoid function (*neuron4*) to *neuron2*. The result of *neuron2* is connected to *recA* by two ways directly under a connection weight and by another interconnection *hid3* and *neuron1* and then *recA*. In other word, *hid1*, *hid3* can transmit their internal data under different delays to *recA*. Moreover, *hid2* does not have any inputs but its value has effect on *recA* through *neuron1* and *neuron3*. Therefore *hid2* shows the effect of an unknown hidden variable which is effective on the target gene *recA*.

Results illustrate the capability of PANNET to model the temporal gene expression levels and reveal the regulatory pathways of three target genes in E. coli. The proposed approach is unique in that it allowed us to consider the effect of hidden/delayed interactions between different components of the network.

a.



b.



c.

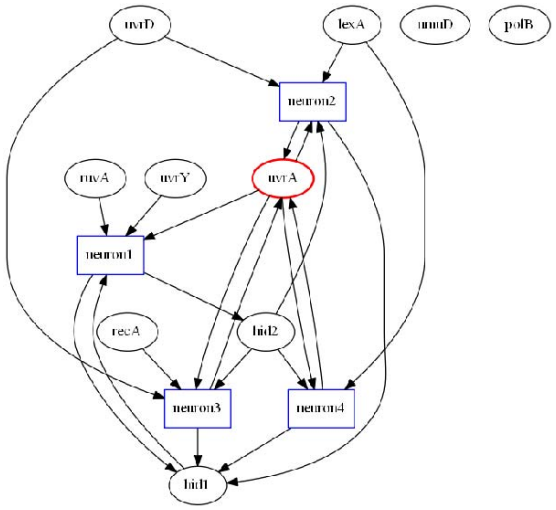


Fig. 7. Constructed models of regulations of target gene a. *lexA*, b. *recA*, c. *uvrA*

IV. Conclution

An important concern in modeling gene regulatory interactions is that the underlying biological processes may take place at different time points, and thus, all of genetic interactions might be delayed. Moreover, regulatory intermediate activities (hidden variables), such as microRNAs, proteins, and metabolites, which are not always represented in microarray experiments, may affect gene interactions. Considering the effect of different time delays and unknown hidden variables on gene expression levels are critical issues and must be taken into account during mathematical modeling. To this aim, we developed an evolvable partially connected artificial neural network to generate a nonlinear model in which some extra nodes are added to the structure of the network to mimic the role of memory or internal states of the system that are not already captured. The interactions between GRN components are determined while encompassing the effects of unknown hidden variables and different time delays. The proposed method improved the GRN modeling, and allowed us to understand the dynamical nature of fundamental biological processes using a nonlinear sparse configuration.

Acknowledgment

The authors acknowledge Dr. David Carroll, for providing feedback to this research whose contribution is supported by the National Center for Advancing Translational Sciences (NCATS), National Institutes of Health, through Grant Award Number 1UL1TR001111. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

[1] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo, "How to infer gene networks from expression profiles," Mol. Syst. Biol., vol. 3, p. 78, Feb. 2007.

[2] M. M. Kordmahalleh, A. Homaifar, and B. K. C. Dukka, "Hierarchical multi-label gene function prediction using adaptive mutation in

crowding niching," in 2013 IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE), 2013, pp. 1–6.
[3] C. Sima, J. Hua, and S. Jung, "Inference of Gene Regulatory Networks Using Time-Series Data: A Survey," Curr. Genomics, vol. 10, no. 6, pp. 416–429, Sep. 2009.
[4] Z. Wang, F. Yang, D. W. C. Ho, S. Swift, A. Tucker, and X. Liu, "Stochastic Dynamic Modeling of Short Gene Expression Time-Series Data," IEEE Trans. NanoBioscience, vol. 7, no. 1, pp. 44–55, Mar. 2008.
[5] A. R. Chowdhury, M. Chetty, and N. X. Vinh, "Incorporating time-delays in S-System model for reverse engineering genetic networks," BMC Bioinformatics, vol. 14, no. 1, p. 196, Jun. 2013.
[6] M. Zou and S. D. Conzen, "A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data," Bioinformatics, vol. 21, no. 1, pp. 71–79, Jan. 2005.
[7] M. Quach, N. Brunel, and F. d' Alché-Buc, "Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference," Bioinformatics, vol. 23, no. 23, pp. 3209–3216, Dec. 2007.
[8] I. Pournara and L. Wernisch, "Factor analysis for gene regulatory networks and transcription factor activity profiles," BMC Bioinformatics, vol. 8, p. 61, Feb. 2007.
[9] K. S. Rainer Opgen-Rhein, "Inferring gene dependency networks from genomic longitudinal data: a functional data approach," REVSTAT – Stat. J. Vol., vol. 4, no. 1, pp. 53–65, 2006.
[10] A. de la Fuente, N. Bing, I. Hoeschele, and P. Mendes, "Discovery of meaningful associations in genomic data using partial correlation coefficients," Bioinforma. Oxf. Engl., vol. 20, no. 18, pp. 3565–3574, Dec. 2004.
[11] A. Wille, P. Zimmermann, E. Vranova, A. Furholz, O. Laule, S. Bleuler, L. Hennig, A. Prelic, P. von Rohr, L. Thiele, E. Zitzler, W. Gruissem, and P. Buhlmann, "Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana," Genome Biol., vol. 5, no. 11, p. R92, 2004.
[12] E. P. van Someren, B. L. T. Vaes, W. T. Steegenga, A. M. Sijbers, K. J. Dechering, and M. J. T. Reinders, "Least absolute regression network analysis of the murine osteoblast differentiation network," Bioinforma. Oxf. Engl., vol. 22, no. 4, pp. 477–484, Feb. 2006.
[13] D. Bratsun, D. Volfson, L. S. Tsimring, and J. Hasty, "Delay-induced stochastic oscillations in gene regulation," Proc. Natl. Acad. Sci. U. S. A., vol. 102, no. 41, pp. 14593–14598, Oct. 2005.
[14] M. Barrio, K. Burrage, A. Leier, and T. Tian, "Oscillatory Regulation of Hes1: Discrete Stochastic Delay Modelling and Simulation," PLoS Comput Biol, vol. 2, no. 9, p. e117, Sep. 2006.
[15] I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. di Bernardo, and M. P. Cosma, "A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches," Cell, vol. 137, no. 1, pp. 172–181, Apr. 2009.
[16] S. Kim, J. Kim, and K. H. Cho, "Inferring gene regulatory networks from temporal expression profiles under time-delay and noise," Comput. Biol. Chem., vol. 31, no. 4, pp. 239–245, Aug. 2007.
[17] R. Xu, D. Wunsch Ii, and R. Frank, "Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization," IEEEACM Trans. Comput. Biol. Bioinforma. IEEE ACM, vol. 4, no. 4, pp. 681–692, Dec. 2007.
[18] X. Hu, A. Maglia, and D. Wunsch, "A general recurrent neural network approach to model genetic regulatory networks," Conf. Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Conf., vol. 5, pp. 4735–4738, 2005.
[19] N. Noman, L. Palafox, and H. Iba, "Reconstruction of Gene Regulatory Networks from Gene Expression Data Using Decoupled Recurrent Neural Network Model," in Natural Computing and Beyond, Springer Japan, 2013, pp. 93–103.
[20] O. ElBakry, M. O. Ahmad, and M. N. S. Swamy, "Inference of Gene Regulatory Networks with Variable Time Delay from Time-Series Microarray Data," IEEE/ACM Trans. Comput. Biol. Bioinform., vol. 10, no. 3, pp. 671–687, May 2013.
[21] A. Lotsi and E. Wit, "State-space modeling of dynamic genetic networks," ArXiv13083590 Stat, Aug. 2013.
[22] M. Moradi Kordmahalleh, M. Gorji Sefidmazgi, A. Homaifar, D. B. KC, and A. Guiseppi-Elie, "Time-series Forecasting with Evolvable Partially Connected Artificial Neural Network," in Proceedings of the 2014 Conference Companion on Genetic and Evolutionary Computation Companion, 2014, pp. 79–80.
[23] S. O. Haykin, Neural Networks and Learning Machines, 3 edition. New York: Prentice Hall, 2008.
[24] M. Gorji Sefidmazgi and M. Farrokhi, "Online Calibration of Inertial Sensors Using Kalman Filters and Artificial Neural Networks," presented at the Iranian Conference on Electrical Engineering, 2009, pp. 249–253.
[25] B. Ristevski, "A survey of models for inference of gene regulatory networks," Nonlinear Anal. Model. Control, vol. 18, no. 4, pp. 444–465, 2013.
[26] D. Elizondo and E. Fiesler, "A Survey of Partially Connected Neural Networks," Int. J. Neural Syst., vol. 08, no. 05n06, pp. 535–558, Oct. 1997.
[27] R. Xu, G. K. Venayagamoorthy, and D. C. Wunsch II, "Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization," Neural Netw., vol. 20, no. 8, pp. 917–927, Oct. 2007.
[28] J. Riquelme, M. A. Ridao, E. F. Camacho, and M. Toro, "Using Genetic Algorithms with Variable-length Individuals for Planning Two-Manipulators Motion," in Artificial Neural Nets and Genetic Algorithms, Springer Vienna, 1998, pp. 26–30.
[29] H. Xiong and Y. Choe, "Structural systems identification of genetic regulatory networks," Bioinformatics, vol. 24, no. 4, pp. 553–560, Feb. 2008.
[30] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics," Proc. Natl. Acad. Sci. U. S. A., vol. 99, no. 16, pp. 10555–10560, Aug. 2002.
[31] R. Manshaei, P. S. Bidari, J. Alirezaie, and M. A. Malboobi, "Time Series Prediction of Gene Expression in the SOS DNA Repair Network of Escherichia coli Bacterium Using Neuro-Fuzzy Networks," in 13th International Conference on Biomedical Engineering, C. T. Lim and J. C. H. Goh, Eds. Springer Berlin Heidelberg, 2009, pp. 1846–1849.
[32] D.Y. Cho, K.-H. Cho, and B.-T. Zhang, "Identification of biochemical networks by S-tree based genetic programming," Bioinforma. Oxf. Engl., vol. 22, no. 13, pp. 1631–1640, Jul. 2006.
[33] S. Kimura, K. Sonoda, S. Yamane, H. Maeda, K. Matsumura, and M. Hatakeyama, "Function approximation approach to the inference of reduced NGnet models of genetic networks," BMC Bioinformatics, vol. 9, p. 23, 2008.
[34] M. Kabir, N. Noman, and H. Iba, "Reverse engineering gene regulatory network from microarray data using linear time-variant model," BMC Bioinformatics, vol. 11, no. Suppl 1, p. S56, Jan. 2010.
[35] S. Kimura, S. Nakayama, and M. Hatakeyama, "Genetic network inference as a series of discrimination tasks," Bioinforma. Oxf. Engl., vol. 25, no. 7, pp. 918–925, Apr. 2009.
[36] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d' Alché–Buc, "Gene networks inference using dynamic Bayesian networks," Bioinformatics, vol. 19, no. suppl 2, pp. ii138–ii148, Sep. 2003.