01 Jan 2005

A General Recurrent Neural Network Approach to Model Genetic Regulatory Networks

Xiao Hu

Anne M. Maglia
Missouri University of Science and Technology

Donald C. Wunsch
Missouri University of Science and Technology, dwunsch@mst.edu

Recommended Citation

X. Hu et al., "A General Recurrent Neural Network Approach to Model Genetic Regulatory Networks," *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS (2005, Shanghai, China)*, pp. 4735-4738, Institute of Electrical and Electronics Engineers (IEEE), Jan 2005.
The definitive version is available at https://doi.org/10.1109/IEMBS.2005.1615529

Proceedings of the 2005 IEEE
Engineering in Medicine and Biology 27th Annual Conference
Shanghai, China, September 1-4, 2005

A General Recurrent Neural Network Approach to Model Genetic Regulatory Networks

Xiao Hu[1], Ann Maglia[2], Donald C. Wunsch II[1]

[1]Applied Computational Intelligence Lab
Department of Electrical & Computer Engineering
University of Missouri-Rolla
Rolla, MO 65409 USA
Email: {xhu; dwunsch}@umr.edu

[2]Department of Biology and Sciences
University of Missouri-Rolla
Rolla, MO 65409 USA
Email: magliaa@umr.edu

Abstract -**There is an urgent need for tools to unravel the complex interactions and functionalities of genes. As such, there has been much interest in reverse-engineering genetic regulatory networks from time series gene expression data. We use an artificial neural network to model the dynamics of complicated gene networks and to learn their parameters. The positive and negative regulations of genes are defined by a weight matrix, and different genes are allowed to have different decaying time constants. We demonstrate the effectiveness of the method by recreating the *SOS DNA Repair* network of *Escherichia coli* bacterium, previously discovered through experimental data.**

I. INTRODUCTION

With the sequencing of the human genome [1], tools are urgently needed to unravel the interactions and functionalities of genes. Fortunately, microarray technology provides the opportunity to perform large-scale gene expression analyses [2]. However, the overwhelming complexity of signal transduction pathways, and the cascade of information sent from the plasma membrane to the gene, hinder the understanding of the mechanisms and control of gene expression. This is further exacerbated, because gene expression is controlled by many elements, including the presence of gene-specific transcription factors.

Fortunately, as additional gene expression data are collected, the modeling of gene expression pathways is becoming a reality. Currently, the most popular approach to modeling gene expression, and a major area in the field of bioinformatics, is to develop gene networks. Gene networks can be thought of simply as logical networks of nodes that influence each other's expression levels. Using microarray expression data, gene networks can be reverse-engineered to describe the expression and control pathways.

Several methodologies have been proposed for constructing genetic network inference based on gene expression or protein data, including Boolean networks [3][4], linear differential models [5][6] and Bayesian networks [7][8]. However, finding a model that successfully infers a genetic network has been elusive for many reasons [9].

Herein, we propose an alternative approach to reverse-engineering gene networks, namely Recurrent Neural Network (RNN). The motivation for exploring RNN's architectures is its potential for dealing with temporal behavior. Recurrent network is capable of settling on a solution, such as in a vision system, that gradually solves a complex set of conflicting constraints to arrive at an interpretation [10][11][12]. In using RNN for genetic network inference, we are mainly concerned with the ability of RNN to interpret complex temporal behavior. By using RNN, we are able to overcome the unrealistic properties of linear model by introducing a nonlinear transfer function (similar to a dose-response curve), and an explicit mRNA decay term. Generalized recurrent neural network model can be considered as signal processing units forming a global regulatory network. The most relevant work to ours, so far, has been [13][14][15]. Our work is distinct from these because we report specific results on both weight matrix and decay time constant learning.

The remainder of this paper is organized as follows. In the next section, we will describe the RNN model. We will then discuss the results of the experimental data related to the *SOS DNA Repair* network of *Escherichia coli* bacterium. Finally, we present the conclusions.

II. MODEL

The proposed model is based on the assumption that the regulatory effect of genes can be expressed as a neural network, wherein nodes represent genes and the connections between nodes define regulatory interactions. We consider the most common neural network formulation,

$$T_i \frac{dy_i}{dt} = -y_i + \sigma(x_i) + I_i , \qquad (1)$$

where y_i is the state or activation level of node i, and

$$x_i = \sum_j w_{ji} y_j + b_i \qquad (2)$$

is the total input to node i; w_{ij} is the connection from node i to j; b_i is the bias term; $\sigma(\xi) = (1+e^{-\xi})^{-1}$. The initial conditions $y_i(t_0)$ and driving functions $I_i(t)$ are the inputs to the system. Since the output of each node has a connec-

tion back to its input, this model is generally considered a recurrent network. It can represent positive and negative regulatory effects between genes by having positive or negative connections. The production term and decaying term also have implications in the biological process. External inputs, $I_i(t)$, represent additional stimuli outside of the genetic network, such as effects from regulatory elements, related protein levels, and other genes that are not in the network. We can develop a more general format of Equation (2) by introducing time delays between the output of a gene and its effect on another, as:

$$x_i = \sum_j w_{ji} y_j(t - \tau_{ji}) + b_i, \quad (3)$$

where τ_{ji} represents the time delay between the output of gene *j* and its effect on gene *i*. This could correspond to time delays incurred during the transcription and translation steps of gene expression, possible transport of the generated protein through the cell, or simply the effect of a series of intermediate steps that are not explicitly being modeled as nodes in the network [13]. As a result, the learning rule is generalized and this general model more closely approximates biological reality. Thus, the recurrent neural network is similar to the genetic regulatory network. It is an obvious advantage to infer a genetic network with a model that shares common properties with it.

The model is shown in Fig. 1 and is derived by discretizing Equation (1), as:

$$y_i(t + \Delta t) = (1 - \frac{\Delta t}{T_i}) y_i(t) + \frac{\Delta t}{T_i} \sigma(x_i(t)) + \frac{\Delta t}{T_i} I_i(t). \quad (4)$$

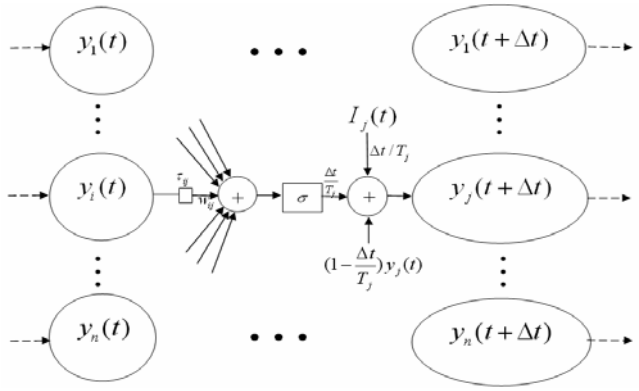

Fig. 1. A recurrent network model of a gene regulatory network, described by Equation (4). It depicts the evolvement of gene expression level of genes in the network.

Given the gene expression measurement in a time series, the objective is to recover the genetic network, which behaves in a manner reflected in the collected data. In other words, the regulatory interactions strength, w_{ij}, and node parameters, T_i, are to be recovered. Details of the learning algorithms are available in [16]. To add to the completeness of our description, we present the learning rule in a continuous-time format with concise notes.

Consider minimizing $E(y)$, some function of the trajectory taken by y between t_0 and t_1. For instance:

$$E = \sum_{t_0}^{t_1} \sum_{i=1}^{n} (y_i(t) - d_i(t))^2, \quad (5)$$

where $d(t)$ is the measured expression level of gene *i* at time *t* and *n* is the total number of genes in the measurement. More elaborate error terms can be easily added. Backpropagation Through Time (BPTT) [17] is used to find the derivatives of an error term *E* with respect to the individual weights w_{ij} of the network, updating them in the direction that minimizes *E*.

The learning rule in the continuous-time form, by taking the limit as $\Delta t \to 0$ is:

$$\frac{dz_i(t)}{dt} = \frac{1}{T_i} z_i(t) - e_i(t) - \sum_j \frac{1}{T_j} w_{ij} \sigma'(x_j(t + \tau_{ij})) z_j(t + \tau_{ij}), \quad (6)$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{1}{T_j} \int_{t_0}^{t_1} y_i(t) \sigma'(x_j(t + \tau_{ij})) z_j(t + \tau_{ij}) dt, \quad (7)$$

$$\frac{\partial E}{\partial b_i} = \frac{1}{T_i} \int_{t_0}^{t_1} \sigma'(x_i(t)) z_i(t) dt, \quad (8)$$

$$\frac{\partial E}{\partial T_i} = -\frac{1}{T_i} \int_{t_0}^{t_1} z_i(t) \frac{dy_i(t)}{dt} dt, \quad (9)$$

where ∂^+ denotes the ordered derivative [17].

Instead of regarding the time delays as a fixed part of the architecture, we can also consider them as tunable parameters. We could learn the time delays in the continuous-time format:

$$\frac{\partial E}{\partial \tau_{ij}} = -\frac{1}{T_j} \int_{t_0}^{t_1} z_j(t) \sigma'(x_j(t)) w_{ij} \frac{dy_i(t - \tau_{ij})}{dt} dt, \quad (10)$$

or in the discrete format:

$$\frac{\partial E}{\partial \tau_{ij}} = -\frac{1}{T_j} \sum_{t=t_0}^{t_1} z_j(t + \Delta t) \sigma'(x_j(t) w_{ij} (y_i(t + \Delta t - \tau_{ij}) - y_i(t - \tau_{ij})). \quad (11)$$

We use the learning rule described above to learn the weight matrix and the time constants associated with each gene from a public known biological data set. The most relevant work to ours, so far, has been [13][14][15]. D'haeseleer [13] conducted preliminary exploration on the use of dynamic recurrent neural networks to restore a sparse genetic network from simulated data, but the network was restricted only by learning w_{ij}, and no time constant T_i learning or time delays τ_{ij} tuning was involved. The w_{ij} values had significant discrepancies, due to limited data. Vohradský [16] also discussed the biological plausibility of the neural network model of gene expression. No attempt was made to learn w_{ij} and T_i. Our work is distinct from these because we report specific results on both weight matrix and decay time constant learning.

III. EXPERIMENTAL RESULTS

To determine if the model is as efficient in predicting real biological data, we test our methods using the gene regulatory network published by Ronen et al. [18] for the *SOS*

DNA Repair network of the *Escherichia coli* bacterium (Fig. 2).

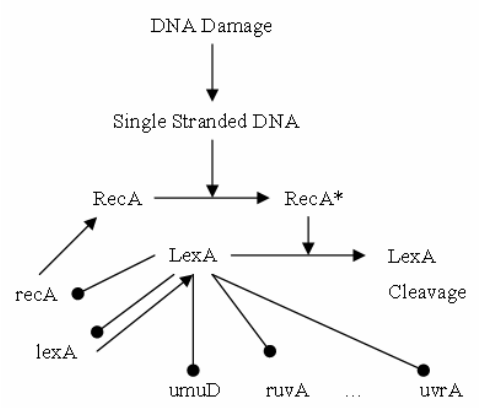

Fig. 2. An example of genetic regulatory network - the SOS DNA repair network. Inhibitions are represented by –●, while activations are represented by →.

Although four experiments were conducted by [18] under various light intensities (Exp. 1&2: 5 Jm^{-2}, Exp. 3&4: 20 Jm^{-2}), we use only the data for experimental condition 1. Data are expression kinetics of the eight main genes of the SOS network. Each experiment is composed of fifty time points, sampled every six minutes, and eight major genes are monitored: uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA and polB. All data collected by [18] are available for download at Uri Alon's homepage (http://www.weizmann.ac.il/mcb/UriAlon/Papers/SOSData/).

We proceed to several learning experiments on the data provided by Ronen *et al.*. Fig. 3 shows the plots of the real gene expression profile and the learned profile from the experimental data after averaging the results of thirty simulations. The trends of the most-expressed genes are fairly well modeled. It is important to keep in mind that the goal of genetic network inference is to recover the regulatory interaction weight matrix. In principle, a positive connection w_{ij} indicates that gene *j* activates gene *i*, and a negative weight w_{ij} represents that gene *j* inhibits gene *i.* Normally, when the weight coefficient w_{ij} is zero or nearly zero, we assume that gene *j* has no regulatory effect on gene *i*. We use the same approach as in [28] to identify regulatory relationships so that we could compare our results with those from dynamic Bayesian network models [28]. The learned values of each parameter w_{ij} are distributed with mean μ_{ij} and variance σ_{ij}^2 in the thirty simulations. The mean and variance of all 64 coefficients, μ and σ^2, also are computed. Weight coefficients are then discretized into four classes according to their mean and standard deviation:

- Class[+] : $\mu_{ij} > \mu + \sigma$ and $\sigma_{ij} < |\mu_{ij}|$,
- Class[-] : $\mu_{ij} < \mu \bullet \sigma$ and $\sigma_{ij} < |\mu_{ij}|$,
- Class[0] : $|\mu_{ij}| < \sigma$ and $\sigma_{ij} < \sigma$,
- Class[X] : other coefficients.

(a)

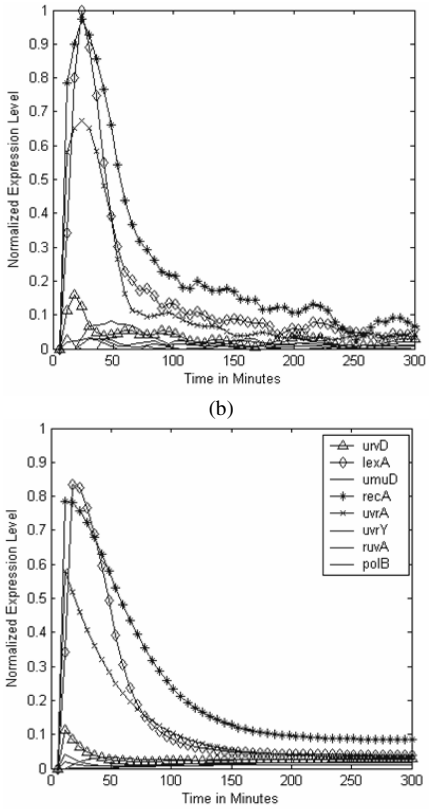

Fig. 3. The comparison of the dynamic behavior in terms of gene expression level between the learned network and the real genetic network. (a) Real gene expression profile under the experimental condition 1 – UV light dose: 5 Jm^{-2};(b) Learned gene expression profile from the data under the experimental condition 1 after averaging the results of 30 simulations.

Classes are built to represent, respectively, probable activations (Class[+]), probable inhibitions (Class[-]), probable absence of regulation (Class[0]), and probable presence of unknown regulations (Class[X]). The network was simulated thirty times under different configurations. Fig. 4 shows the identified discretized regulatory matrix *W*.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \pm & 0 & + & 0 & 0 & 0 & 0 \\ 0 & - & 0 & + & 0 & 0 & 0 & 0 \\ 0 & \pm & 0 & + & X & X & 0 & 0 \\ 0 & \pm & 0 & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & + & 0 & 0 & 0 & 0 \\ 0 & - & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & - & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} uvrD \\ lexA \\ umuD \\ recA \\ uvrA \\ uvrY \\ ruvA \\ polB \end{matrix}$$

Fig. 4. Weight identified after thirty simulations. The *j*th column shows all identified regulations inflicted by *j*th gene on the other genes. Inversely, the *i*th row shows all regulations the *i*th gene is submitted to. Genes are listed on the right.

Compared to the nine probable regulations identified in [28], our recurrent neural network model identifies seven of them with an additional six probable regulations. The results show that we can identify the inhibition of *LexA* on *umuD*, *ruvA*, and *polB*, and the activation of *recA* on *lexA*, *umuD*, *recA* and *uvrY*. The regulations of *LexA* on *lexA*,

recA and *uvrA* are likely because they are always identified in the results. These discretized weights are marked with ± in Fig. 4 because both + and • are identified in the results under different regularization parameter configurations. When conflicts such as these occur, it may be necessary to investigate further those particular regulatory relationships through laboratory experimentation.

Another interesting result from our experiments is the learned time constants of the genes. Fig. 5 depicts the mean and standard derivation of the learned time constants.

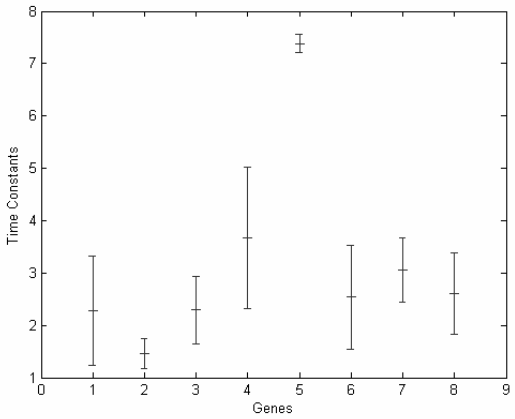

Fig. 5. The learned time constants (from thirty simulatins) vs. genes. 1 – *uvrD*, 2 – *lexA*, 3 – *umuD*, 4 – *recA*, 5 – *uvrA*, 6 – *uvrY*, 7 – *ruvA*, 8 – *polB*. Error bars indicate their mean and standard deviation.

As shown in Fig. 5, gene *uvrA* always has a relatively large time constant, while the gene *lexA* always has a relatively small time constant, in all simulations, compared to the other genes. These inferred time constants give us hints of the real decaying rate (as represented by T_i in Equation (1)) of the gene expression level under certain environmental conditions, and might be used to explain the kinetics of all *SOS* genes. Furthermore, they can provide initial evidence that can be used to design experiments to explore other regulations between inside and outside of the network.

IV. CONCLUSIONS AND DISCUSSIONS

Although the use of recurrent neural network models to reverse-engineering gene networks has been proposed by others, to date, the design and simulation of recurrent network models have not been discussed in details or simulated using real biological data. In this paper, we use the general recurrent neural network to model the dynamics of complicated gene networks and to learn their parameters. At the learning of the real biological data set – the data from the SOS DNA Repair Network of the E. Coli bacterium, the model is able to discover more complex regulation relationships among genes in the SOS network, compared to the results from the dynamic Bayesian network model. At the same time, with a learning regulatory matrix, the RNN model can further infer the gene decaying rate T_i, which can be used to explain the kinetics of the genes. It is also highly adaptable, in that it can accommodate the addition of time delay factors τ_{ij} and other applicable error terms into the model.

Given the similarity between recurrent neural network and gene regulatory network, RNN should play an important role in unraveling the mystery of gene regulation relationships and their roles in controlling developmental, physiological and pathological processes.

ACKNOWLEDGMENT

Support for this research from the National Science Foundation, and from the M.K. Finley Missouri endowment, is gratefully acknowledged.

REFERENCES

[1] B. R. Jasny, L. Roberts, Building on the DNA Revolution. *Science* 300: 277, 2003.
[2] L. F. A. Wessels, E. P. Van Someren, M.J.T. Reinders, "A comparison of genetic network models," Pac Symp Biocomput, 508-519, 2001.
[3] T. Akutsu, S. Miyano, S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," Pac Symp Biocomput, 17-28, 1999.
[4] S. Liang, S. Fuhrman, R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," Pac Symp Biocomput, 18-29, 1998.
[5] J. L. Michael De Hoon, S. Imota, K. Kobayashi, N. Ogasawara, S. Miyano, "Inferring gene regulatory networks from time-ordered gene expression data of Bacillus subtills using differential equations," Pac Symp Biocomput, 17-28, 2003.
[6] T. Chen, H. L. He, G. M. Church, "Modeling gene expression with differential equations," Pac Symp Biocomput, 4:29-40, 1999.
[7] N. Friedman, M. Linial, I. Nachman, D. Pe'er, "Using Bayesian net work to analyze expression data," J. Comp. Biol., 7, 601-620, 2000.
[8] S. Imoto, T. Gota, S. Miyano, "Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression," Pac Symp Biocomput, 175-186, 2002.
[9] R. Xu, X. Hu, D. Wunsch, "Inference of genetic regulatory networks with recurrent neural network models," Engineering in Medicine and Biology Society, 2004. EMBC 2004. Conference Proceedings. 26th Annual International Conference of the Volume 2, 1-5 Sept. 2004 Page(s):2905 –2908 Vol.4.
[10] J. F. Kolen, S. C. Kremer, A Field Guide to Dynamic Recurrent Networks, Wiley-IEEE Press, New York, March 2001.
[11] F. J. Pineda. Generalization of back-propagation to recurrent neural networks. Physical Review Letters 59, 19, 2229-2232, 1987.
[12] B. A. Pearlmutter, "Learning state space trajectories in recurrent neural networks," Neural Computation 1, 2, 263-269, 1989.
[13] P. D'haeseleer, Reconstructing gene networks from large scale gene expression data. Ph.D. Thesis, 2000.
[14] E. Mjolsness, D. H. Sharp, J. Reinitz, "A Connectionist Model of Development." J. Theoretical Biology 152, 429-453, 1991.
[15] Jiří Vohradský, "Neural network model," J Biol Chem 276, 39, 36168-36173, 2001.
[14] B. A. Pearlmutter, "Gradient calculations for dynamic recurrent neural networks: a survey," IEEE Transactions on Neural Networks 6(5), 1212-1218, 1995.
[15] P. J. Werbos, "Backpropagation through time: what it does and how to do it," Proceedings of IEEE, 78(10), 1550-1560, 1990.
[18] M. Ronen, R. Rosenberg, B. I. Shraiman, U. Alon, "Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics," Proc. Natl Acad. Sci. USA, 99, 10555-10560.