A general recurrent neural network approach to model genetic regulatory networks

Xiao Hu

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

Anne M. Maglia  
*University of Missouri--Rolla*

Follow this and additional works at: [http://scholarsmine.mst.edu/faculty_work](http://scholarsmine.mst.edu/faculty_work)

Part of the [Biology Commons](http://scholarsmine.mst.edu/faculty_work/651), and the [Electrical and Computer Engineering Commons](http://scholarsmine.mst.edu/faculty_work/651)

Recommended Citation


[http://scholarsmine.mst.edu/faculty_work/651](http://scholarsmine.mst.edu/faculty_work/651)

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. For more information, please contact weaverjr@mst.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 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,

\[
T_i \frac{dy_i}{dt} = -y_i + \sigma(x_0) + I_i ,
\]

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

\[
x_i = \sum_j w_{ij} y_j + b_i \tag{2}
\]

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:

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

where $\tau_{ij}$ 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:

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

### 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 \( - \) \( \gamma \), 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 \( \mu_j \) and variance \( \sigma_j^2 \) in the thirty simulations. The mean and variance of all 64 coefficients, \( \mu \) and \( \sigma^2 \), also are computed. Weight coefficients are then discretized into four classes according to their mean and standard deviation:

- Class[+] : \( \mu_j > \mu + \sigma \) and \( \sigma_j < |\mu_j| \),
- Class[-] : \( \mu_j < \mu - \sigma \) and \( \sigma_j < |\mu_j| \),
- Class[0] : \( |\mu_j| < \sigma \) and \( \sigma_j < \sigma \),
- Class[X] : other coefficients.

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 \( W \).

![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.](image)

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 \( \text{LexA} \) on \( \text{umuD} \), \( \text{ruvA} \), and \( \text{polB} \), and the activation of \( \text{recA} \) on \( \text{lexA} \), \( \text{umuD} \), \( \text{recA} \) and \( \text{uvrY} \). The regulations of \( \text{LexA} \) on \( \text{lexA} \),
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


\( \text{recA} \) and \( \text{uvrA} \) 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.

As shown in Fig. 5, gene \( \text{uvrA} \) 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 \( \text{SOS} \) 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 \( \tau_i \) and other applicable error terms into the model.