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
Part of the Biology Commons, and the Electrical and Computer Engineering Commons

Recommended Citation
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.
A General Recurrent Neural Network Approach to Model Genetic Regulatory Networks

Xiao Hu¹, Ann Maglia², Donald C. Wunsch II¹

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

²Department 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,

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

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

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

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:

\[
x_i = \sum_j w_{ij} y_j(t-\tau_j) + b_i,
\]

where \( \tau_j \) 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:

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

![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.](image)

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 \( t_f \). For instance:

\[
E = \sum_{i=1}^{n} \sum_{t=1}^{T} (y_i(t) - d_i(t))^2,
\]

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:

\[
\frac{dz_i(t)}{dt} = \frac{1}{T_i} z_i(t) - e_i(t) - \sum_j \frac{1}{T_j} w_{ij} \sigma(x_j(t+\tau_j)) z_j(t+\tau_j),
\]

\[
\frac{\partial E}{\partial w_{ij}} = \frac{1}{T_i} \int_0^1 y_j(t) \sigma(x_j(t+\tau_j)) z_j(t+\tau_j) dt,
\]

\[
\frac{\partial E}{\partial b_i} = \frac{1}{T_i} \int_0^1 \sigma(x_i(t)) z_i(t) dt,
\]

\[
\frac{\partial E}{\partial T_i} = -\frac{1}{T_i} \int_0^1 z_i(t) \frac{dy_i(t)}{dt} dt,
\]

where derivative 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:

\[
\frac{\partial E}{\partial \tau_{ij}} = -\frac{1}{T_j} \int_0^1 z_j(t) \sigma(x_j(t)) w_{ij} \frac{dy_i(t-\tau_{ij})}{dt} dt,
\]

or in the discrete format:

\[
\frac{\partial E}{\partial \tau_{ij}} = -\frac{1}{T_j} \sum_{t=1}^{T} z_j(t+\Delta t) \sigma(x_j(t)) w_{ij} (y_i(t+\Delta t-\tau_{ij}) - y_i(t-\tau_{ij})).
\]

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 \( \tau_{ij} \) tuning was involved. The \( w_{ij} \) values had significant discrepancies, due to limited data. Vohradsky [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).

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 \( ijw \) indicates that gene \( j \) activates gene \( i \), and a negative weight \( ijw \) 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 \( \mu_{ij} \) and variance \( \sigma^2_{ij} \) 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[+] : \( \mu_{ij} > \mu + \sigma \) and \( \sigma_{ij} < |\mu_{ij}| \),
- Class[-] : \( \mu_{ij} < \mu - \sigma \) and \( \sigma_{ij} < |\mu_{ij}| \),
- Class[0] : \( |\mu_{ij}| < \sigma \) and \( \sigma_{ij} < \sigma \),
- Class[X] : other coefficients.

\[ \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

\( uvrD \)

\[ \begin{pmatrix}
0 \pm 0 + 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

\( lexA \)

\[ \begin{pmatrix}
0 - 0 + 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

\( umuD \)

\[ \begin{pmatrix}
0 \pm 0 + X & X & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

\( recA \)

\[ \begin{pmatrix}
0 \pm 0 X & X & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

\( uvrA \)

\[ \begin{pmatrix}
0 \pm 0 + 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

\( uvrY \)

\[ \begin{pmatrix}
0 - 0 + 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

\( ruvA \)

\[ \begin{pmatrix}
0 - 0 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

\( polB \)

\[ \begin{pmatrix}
0 - 0 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]

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*,...
 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