Inference of genetic regulatory networks from time series gene expression data

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

Rui Xu
University of Missouri--Rolla

Xiao Hu

Follow this and additional works at: http://scholarsmine.mst.edu/faculty_work

Part of the Electrical and Computer Engineering Commons

Recommended Citation
Wunsch, Donald C.; Xu, Rui; and Hu, Xiao, "Inference of genetic regulatory networks from time series gene expression data" (2004). Faculty Research & Creative Works. Paper 1717.
http://scholarsmine.mst.edu/faculty_work/1717

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.
Inference of Genetic Regulatory Networks from Time Series Gene Expression Data

Rui Xu, Xiao Hu, and Donald C. Wunsch II
Applied Computational Intelligence Laboratory
Dept. of Electrical and Computer Engineering
University of Missouri - Rolla
Rolla, MO 65409-0249 USA
rxu@umr.edu, xhu@umr.edu, dwunsch@ece.umr.edu

Abstract—Large-scale gene expression data coming from microarray experiments provide us a new means to reveal fundamental cellular processes, investigate functions of genes, and understand relations and interactions among them. To infer genetic regulatory networks from these data with effective computational tools has become increasingly important. Several mathematical models, including Boolean networks, Bayesian networks, linear additive model, and recurrent neural networks, have been used to explore the behaviors of regulatory networks. In this paper, we survey these methods in the inference of genetic regulatory networks from time series gene expression data.

I. INTRODUCTION

With the emergence and rapid advancement of DNA microarray technologies [1, 2], to infer genetic regulatory networks from time series gene expression data has become increasingly important in order to reveal fundamental cellular processes, investigate functions of genes, and understand complex relations and interactions among genes [3, 4]. A genetic regulatory network consists of a set of DNA, RNA, proteins, and other molecules, and describes regulatory mechanisms among these components. Genetic information that determines structures, functions, properties of living cells is stored in DNA, whose coding regions, known as genes, encode proteins. According to the central dogma of molecular biology, genes are transcribed into mRNA molecules, which are then translated to proteins. Since all cells for a specific organism include the same genetic material, it is important to know which proteins are synthesized, or which genes are expressed, under certain conditions. For example, only eye cells synthesize proteins that determine the color of eyes. This is achieved through the actions of some proteins, which activate or inhibit the transcription rate of certain genes by binding to regions either upstream or downstream of these genes. Therefore, the transcription of a specific gene, or the control of its gene expression, can be regarded as a combinatorial effect of a set of other genes. Fig. 1 illustrates the SOS DNA Repair network in bacterium Escherichia coli as a simple example [5]. The SOS system consists of around 30 genes regulated at the transcriptional level. When damage occurs, protein

RecA, which functions as a sensor of DNA damage, becomes activated and mediates LexA autocleavage by binding to single-stranded DNA molecule. Protein LexA is a master repressor that represses all genes when no damage occurs. The drop in LexA expression levels causes the activation of the SOS genes. After the damage is repaired or bypassed, the expression level of RecA falls, which causes the accumulation of LexA. Then, LexA binds sites in the promoter regions of these SOS genes and represses their expression. The cells return to their original states.

Conventional methods can only investigate activities between a pair of genes, or among several genes, which is far from sufficient, for exploring the complicated regulatory mechanisms. DNA microarray technologies provide an effective and efficient way to measure gene expression levels of thousands of genes simultaneously under different conditions, which makes it possible to investigate gene activities from the angle of the whole genome [3]. Currently, there are two major types of microarray technologies based on the nature of the attached DNA (cDNA with length varying from several hundred to thousand bases or oligonucleotides containing 20-30 bases). For cDNA technologies, a microarray consists of a solid substrate to which a large amount of cDNA clones are attached according to some certain order [1]. Fluorescently labeled

Fig. 1. An example of genetic regulatory network - the SOS DNA Repair network [5]. Inhibitions are represented by –, while activations are represented by →.

DNA Damage

Single Stranded DNA

RecA → RecA*

LexA → LexA

recA

lexA

umuD

ruvA

uvrA

RecA, which functions as a sensor of DNA damage, becomes activated and mediates LexA autocleavage by binding to single-stranded DNA molecule. Protein LexA is a master repressor that represses all genes when no damage occurs. The drop in LexA expression levels causes the activation of the SOS genes. After the damage is repaired or bypassed, the expression level of RecA falls, which causes the accumulation of LexA. Then, LexA binds sites in the promoter regions of these SOS genes and represses their expression. The cells return to their original states.

Conventional methods can only investigate activities between a pair of genes, or among several genes, which is far from sufficient, for exploring the complicated regulatory mechanisms. DNA microarray technologies provide an effective and efficient way to measure gene expression levels of thousands of genes simultaneously under different conditions, which makes it possible to investigate gene activities from the angle of the whole genome [3]. Currently, there are two major types of microarray technologies based on the nature of the attached DNA (cDNA with length varying from several hundred to thousand bases or oligonucleotides containing 20-30 bases). For cDNA technologies, a microarray consists of a solid substrate to which a large amount of cDNA clones are attached according to some certain order [1]. Fluorescently labeled
A. Boolean Networks

Genetic regulatory network genes are described through a Boolean function. Given a set of gene activities of a single gene to the inference of the whole network, including Boolean networks (DBN) [I 14-16], linear additive regulation model [14-16], and recurrent neural networks (RNN). Bayesian networks [17-20], and dynamic Bayesian networks (DBN) [11-13], are used to model the sparseness of the genetic regulatory networks.

The paper is organized as follows. Section II investigates four major models for the inference of genetic regulatory networks. Several related topics are discussed in Section III and section IV concludes the paper.

II. COMPUTATIONAL MODELS

A. Boolean Networks

Boolean networks are binary models, which consider a gene has only two states: 1 for active and 0 for inactive. The effect of other genes on the state change of a given gene is described through a Boolean function. Given a set of genes 

\[ E = \{ e_1, \ldots, e_N \} \]

where 

\[ e_i \in \{0,1\}, \quad i = 1, \ldots, N \]

represents the state of the \( i \)th gene and 

\[ N \]

is the number of genes, and a set of Boolean functions 

\[ B = \{ b_1, \ldots, b_N \} \]

the dynamics of the network is completely determined through the discrete equations

\[ e_i(t+1) = b_i(e_1(t), \ldots, e_N(t)), \quad i = 1, \ldots, N. \]

Fig. 2 depicts a simple Boolean network consisting of 3 elements. It is easy to achieve the corresponding Boolean equations for each gene as

\[ e_1(t+1) = e_2(t) \lor e_3(t), \]

\[ e_2(t+1) = e_1(t) \land e_3(t), \]

\[ e_3(t+1) = e_1(t) \land e_2(t). \]

Given an initial state (000) at \( t=0 \), the state will change to (010) at the next time point \( t=dt \) according to the system dynamics described in (2).

Although Boolean networks make it possible to investigate the dynamics of a genetic regulatory system, they ignore the effect of genes at intermediate levels, which is not true in reality.

B. Bayesian Networks

A Bayesian network is a graph model to estimate a complicated multivariate joint probability distribution through local probabilities [10]. Under this framework, a genetic regulatory network is described as a directed acyclic graph, including a set of vertices and edges. The vertices are
related to random variables and represent genes or other
components while the edges capture the conditional
dependence relation and represent the interactions among
genes.

Let's consider a simple Bayesian network as shown in
Fig. 3. Each node represents one gene. Since there is no edge
connecting A and C directly, in other words, the effect of
gene A on gene C has to be mediated through gene B, we
say that A and C are conditionally independent. In addition
to a graph that describes (in)dependencies between
variables (genes), each variable is described as a stochastic
function of its parents. Specifically, we associate with each
variable a conditional probability model that specifies the
probability of X given its parents. So, the network structure
in Fig. 3 implies that joint distribution in the product form:

\[ P(A, B, C, D, E) = P(A)P(B|A, E)P(C|B)P(D|A)P(E) \]  

(3)

Using stochastic models is natural in the gene expression
domain because the processes we want to model are
stochastic and also the measurements of the underlying
biological system are noisy [10].

Bayesian networks are effective in dealing with noise,
incompleteness, and stochastic aspects of gene expression
data. The graph representation makes it convenient to
investigate interactions between the genes. However,
Bayesian networks do not consider dynamical aspects of
gene regulation and leave temporal information unhandled.

Recently, dynamic Bayesian networks (DBN) attract
more attention [11-13]. Since gene interactions vary over
time, any static model will fail to represent the mechanism.
Hence, DBN is better than other static models in the sense
of capturing the temporal behavior. DBN can model
behaviors that emerge temporally, e.g., feedback, and
effective in handling problems like hidden variables, prior
knowledge, and missing data. DBN are similar to Bayesian
networks but they are unrolling over time. Suppose we have
T sets of expression measurements \( e(1), e(2), \ldots, e(T) \)
of \( N \) genes, where \( e(t) = (e_1(t), e_2(t), \ldots, e_N(t)) \) is a \( N \)
dimensional gene expression vector obtained from the
microarray at time \( t \). The learning of a DBN is the process
of determining the best fit for a given data set. The network
to be learned is the transition network, which defines the
dependencies between adjacent sample point \( e(t) \) and \( e(t+1) \).

Many important issues arise when learning the structure and
probabilities of a DBN from the given data. Fortunately,
DBN has been well studied in the Artificial Intelligence (AI)
community before it was considered an effective approach
to infer genetic networks. [21-24] have discussed in details
about how to search for the best network structure and
parameters from the given data set. The disadvantage of
DBN is that DBN cannot scale well to large-scale data sets.

C. Linear Additive Regulation Model

Another important model for formalism and simulation
regulatory networks is known as linear additive regulation
models [14-16]. The expression level of a gene at a certain
time point can be calculated by the weighted sum of the
expression levels of all genes in the network at a previous
time point. For a continuous time system, the models can be
represented as

\[ \frac{de_i}{dt} = \sum_{j=1}^{N} w_{ij} e_j + \sum_{k=1}^{K} \nu_{ik} u_k + \beta_i \]  

(4)

where \( e_i(t) \) is the gene expression level for the \( i \)th gene
\((1 \leq i \leq N, N \) is the number of genes in the system),
\( w_{ij} \) represents the effect of the \( j \)th gene on the \( i \)th gene
\((1 \leq i, j \leq N) \), \( u_k \) is the \( k \)th \((1 \leq k \leq K, K \) is the number
of external variables) external variable, \( \nu_{ik} \) represents the
effect of the \( k \)th external variable on the \( i \)th gene, and \( \beta \) is
the bias term. A negative value of \( w_{ij} \) represents the
inhibition of the \( j \)th gene on the \( i \)th gene, while a positive
value indicates the activation controls. When \( u_k \) is zero,
there is no influence of the \( k \)th gene on the expression change
of the \( i \)th gene. This model can also be described in a
discrete form (for computational convenience, since we
only have measurement at some certain time points):

\[ e_i(t + \Delta t) = \sum_{j=1}^{N} \Delta w_{ij} e_j + \sum_{k=1}^{K} \Delta \nu_{ik} u_k + \beta_i + e_i(t) \].  

(5)

Methods for fitting the linear model to the data are rich in
the statistical and mathematical literature. Generally, linear
regression techniques can be used to estimate the
concerning parameters [16, 25].

D’haeseleer used the model to analyze the data
comprising expression level measurements for 112 genes
during the development of the central nervous system (CNS)
of rats [26]. Since the data set consists of measurements
from two tissue types, spinal cord or hippocampus, a flag
variable is introduced to indicate the difference [14].
Experimental results unravel many meaningful gene
interactions and provide biologists with interesting insights
for further analysis.

Although linear additive regulation can reveal certain
linear relations in the regulatory systems, it lacks the
capability to capture the nonlinear dynamics between gene
regulations. Moreover, many assumptions on the model are
not realistic, e.g., there is no mechanism for keeping the
expression level from negative or divergence, as indicated in
[16].
D. Recurrent Neural Networks

Considering the limitation of the linear additive model, we can modify the equation in (4) with a neural network formulation [17-20],

$$\tau \frac{d e_i}{d t} = f \left( \sum_{j=1}^{N_i} w_i e_j + \sum_{k=1}^{R_i} v_i u_k + \beta_i \right) - \lambda e_i, \quad (6)$$

where $f(\cdot)$ is a nonlinear sigmoidal function (usually, $f(x) = 1/(1+e^{-x})$ is used), $\tau$ is the time constant, and $\lambda$ is the decay rate parameter. Correspondingly, its discrete form is

$$e_i(t+\Delta t) = \frac{\Delta t}{\tau} f \left( \sum_{j=1}^{N_i} w_i e_j(t) + \sum_{k=1}^{R_i} v_i u_k(t) + \beta_i \right) + \left( 1 - \frac{\lambda \Delta t}{\tau} \right) e_i(t). \quad (7)$$

Fig. 4 (a) depicts a recurrent neural network, which is unrolled in time from $t=0$ to $T$ with an interval $\Delta t$, for modeling genetic network, particularly for the SOS DNA repair network described in Fig. 1. Fig. 4 (b) illustrates a node in the recurrent neural network, which realizes the equation in (7).

There exist ample training algorithms for training RNN in the literature. Back-Propagation through time (BPTT) is one of the most widely used algorithms and was first proposed by Paul Werbos [27]. It may be derived by unfolding the temporal operation of the network into a layered feedforward network, the topology of which grows by one layer at every time step [28]. In the problem of gene network inference, the goal is to recover the regulatory interactions $w_{ij}$. Considering minimizing $J(e)$, some cost function of trajectory taken by $e$ between $0$ and $T$, for instance,

$$J(e) = \sum_{t=0}^{T} \sum_{i} \left( e_i(t) - d_i(t) \right)^2 \quad (8)$$

which measures the deviation of network output $e(t)$ from the measurement (target) $d(t)$. More elaborate error terms can be easily added. By using BPTT, we find the derivatives of the cost function $J$ with respect to the individual weights $w_{ij}$ of the network. These derivatives can be used to do gradient descent on the weights, updating them in the direction that minimizes $E$:

$$\Delta w_{ij} = -\eta \frac{J(e)}{w_{ij}}. \quad (9)$$

Since it is usually difficult to have the measurements of the external variables, it is common practice to ignore the term $\sum_{t=1}^{T} v_i u_k(t)$ in the derivation of the learning algorithm. Since the training of the RNN need a large amount of the time series data points, we currently use some synthetic data set generated from the network. The BPTT is the algorithm used in the training to learn the weights of the RNN [29].

Other major training algorithms are based on the evolutionary techniques, namely, genetic algorithm (GA) [30] and particle swarm optimization (PSO) [31]. GA tends to optimize a population of structure, consisting of encoded bit strings, called chromosomes, by using a set of evolutionary operators [30]. An optimization function, called the fitness function, is the standard for evaluating the optimizing degree of the population. GA is particularly useful in evolving neural networks where there exist many local optima and traditional gradient-based search algorithms are easy to get stuck. PSO is based on simulation of social behavior. Different from GA, a random velocity is associated with each potential solution, called a particle, which are considered to "be flown through the problem space" [31]. The basic idea of PSO lies in accelerating each particle towards its previous best solution and the overall best locations in the swarm at each time step. It has been shown that PSO require less computational cost and can achieve faster convergence than conventional Back-Propagation in training feedforward neural networks for approximating a nonlinear function [32]. Wahde and Hertz also used GA to the CNS data set [20, 33]. Parameters are encoded based on a decimal scheme. The fitness function is
defined as $f = \frac{1}{1 + \frac{1}{T} \sum \left( e - e^* \right)^2}$, where $T$ is the number of points, $e$ are the data points, and $e^*$ are the estimated values. Experimental results show that the inferred networks can approximate the system dynamics and find important relations between different clusters of genes. Currently, we are testing PSO on modeling the SOS DNA Repair network [5]. The results will be reported elsewhere [38].

In using RNNs for genetic network inference, we are mainly concerned with the ability of RNNs to interpret complex temporal behavior. Generalized recurrent neural network models can be considered as signal processing units forming a global regulatory network. Given the similarity between recurrent neural network and gene networks, we believe that recurrent neural networks, such as the one we have proposed here, will play an important role in unraveling the mystery of gene regulation relationships.

III. DISCUSSIONS

Although time series gene expression data provide a broader prospect for investigating complex gene regulatory activities, and previous exploration with proposed computational and mathematical models has shown promising results, there still exist many challenges requiring more efforts. We summarize and discuss several important problems on the inference of genetic networks as follows.

1. **Curse of dimensionality** [28]. This is the major obstacle for current exploration [16, 25]. Typically, the gene expression data contain measurements of thousands of genes, but only have a limited number of time points (less than 30). This situation limits the application of many data-driven computational models and makes it very difficult to infer a fully determined large-scale regulatory network. Fortunately, biological knowledge on genetic regulatory networks assumes that a gene is only regulated by a limited number of genes. In other words, the regulatory networks are sparsely connected and most weights values are zeroes. Take the RNN model as an example, the inference of genetic regulatory networks can be regarded as a two-step procedure [20, 34]. The goal of the first step is to unravel the interactions between genes in the system, in other words, to determine which weight values are non-zeroes. It is not necessary that the weight values estimated to be precise at this level, and we are only interested in finding the potential regulatory mechanisms from these time series expression data. Following the results of the first stage, we can further fine-tune these non-zero values, while fixing the weights whose values are estimated to be zeroes. Note the procedure may iterate many times depending on the quality of the achieved networks. Below summarizes current strategies for dimensionality reduction.

   1. **Clustering** [16-17, 20]. Clustering algorithms can be used to generate clusters of co-expressed genes based on their expression profiles. For each cluster, mean time series is calculated for further analysis. The disadvantage of this method is that it only identifies relations between the groups of genes, instead of individual genes, which biologists are more interested in.

   2. **Interpolation** [16, 35]. Interpolation may not sufficiently capture some changes between two time points and "tends to reduce dimensionality problem only marginally regardless of the number of time points added" [36].

   3. **Adding noisy duplicates** [36].

   4. **Thresholding**. Remove the genes whose expression levels do not change within some pre-specified values.

   5. **Strategies for network training** [28].
      - Weight decays;
      - Pruning algorithms.

2. **Time delay**. Time delay is ubiquitous in gene regulatory activities and incorporation of time delay can capture the system dynamics more effectively [37]. For example, we can modify (7) to consider the effect of time delay

   \[ e_i(t + \Delta t) = \frac{\Delta t}{r_{i'}} \sum_{j=1}^{K} w_{ij} e_j(t - r_{ij}) + \sum_{k=1}^{K} \nu_{ik} u_k(t) + \beta_i \]

   \[ + \left( 1 - \frac{\Delta t}{r_{ij}} \right) e_i(t) \]  

   (10),

   where $r_{ij}$ is the time delay for the $j$th gene at the transcriptional activity for the $i$th gene.

3. **Robustness and Redundancy** [25]. Genetic networks are known to be robust to noise. Gene expression levels in the networks will not be affected greatly due to the small changes, caused by noise, in expression levels of some genes. Many genes have same functions and express themselves in a similar way under certain pathways. Proposed computational models should be capable of interpreting these phenomena.

IV. CONCLUSIONS

To understand the gene regulatory mechanisms is one of the central tasks in molecular genetics. Inference and simulation of genetic regulatory networks based on the time series gene expression data from microarray experiments becomes an important and effective way to achieve this goal. Here, we introduce several state-of-the-art computational models for modeling regulatory systems and revealing potential gene interactions. Initial experiments have shown promising results and demonstrate the potential of these methods in regulatory network inference. However, with the
limited data, current research only focuses on the modeling of network from synthetic data, or simulation of small-scale network including only several genes or gene clusters. No attempt has been made to infer large-scale genetic regulatory networks. High quality time series gene expression data with sufficient number of time points is particularly important. In the meantime, further improvement for the current computational models is also required in order to explore gene regulation more effectively.

ACKNOWLEDGMENT

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

REFERENCES