Structural systems identification of genetic regulatory networks

Hao Xiong* and Yoonsuck Choe
Department of Computer Science, Texas A&M University, College Station, TX 77843-3112

ABSTRACT

Motivation: Reverse engineering of genetic regulatory networks from experimental data is the first step toward the modeling of genetic networks. Linear state-space models, also known as linear dynamical models, have been applied to model genetic networks from gene expression time series data, but existing works have not taken into account available structural information. Without structural constraints, estimated models may contradict biological knowledge and estimation methods may over-fit.

Results: In this report, we extended expectation-maximization (EM) algorithms to incorporate prior network structure and to estimate genetic regulatory networks that can track and predict gene expression profiles. We applied our method to synthetic data and to SOS data and showed that our method significantly outperforms the regular EM without structural constraints.

Availability: The Matlab code is available upon request and the SOS data can be downloaded from http://www.weizmann.ac.il/mcb/UriAlon/Papers/SOSData/, courtesy of Uri Alon. Zak’s data is available from his website, http://www.che.udel.edu/systems/people/zak

Contact: hxiong@cs.tamu.edu

Supplementary information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

Genetic regulatory networks, often abbreviated as genetic networks, help us untangle the intricate interactions of multiple genes under different environmental conditions. Recent developments in microarray technologies allow scientists to simultaneously measure expressions of thousands, even tens of thousands of genes, over time. The time course of gene expression data can be used to reconstruct genetic networks. In the past decade, a wide variety of models have been developed to study genetic networks. They include Boolean networks (Akutsu et al., 1999; Kauffman et al., 2003), differential equations (Chen et al., 1999; D’Haeseleer et al., 1999) and dynamic Bayesian networks (Dojer et al., 2006; Friedman et al., 2000; Husmeier, 2003; Murphy and Mian, 1999).

A special subclass of dynamic Bayesian networks (DBN) is linear dynamical systems (LDS) (Beal et al., 2005; Dojer et al., 2006; Li et al., 2006; Pe’er et al., 2001), also known as linear state-space models. LDS assumes that observations depend on unobservable states that evolve under Markovian dynamics, i.e. future states are probabilistically determined only by the current state. The state-space approach can provide a general framework for the design of genetic networks in synthetic biology; however, to date, most efforts in estimating an LDS have tried to estimate parameters without considering structural constraints.

Wu et al. (2004) proposed to use LDS to explore large time-course data. While they used hidden, unobserved variables as the states and modeled expression profiles as the output, they did not consider noise, and they estimated states from the output using maximum likelihood factor analysis. The number of states in estimated models was a variable estimated using the Bayesian information criterion (BIC), and the state transition matrix was estimated using least square methods. Wu et al. (2004) considered the state transition matrix as the key parameter that embodies genetic interaction, and the stability or instability of this matrix was cited as supporting evidence for their method. One limitation of Wu’s approach is that factors have no obvious biological counterpart.

Later, Yamaguchi and colleagues (Yamaguchi and Higuchi, 2006; Yamaguchi et al., 2007) also tried to use similar linear state-space system to model genetic networks, where a state was defined as a module of interacting genes. The parameters in the model measured quantitative relationships between modules. The dimension of the states was determined by BIC and the parameters were estimated by the expectation-maximization (EM) algorithm.

Rangel and her colleagues, on the other hand, in a series of papers (Rangel et al., 2001; Rangel et al., 2004a, b), modeled individual gene interactions using linear state-space models. Their estimation method consisted of two parts. The inner part was the EM algorithm with full connectivity assumed and therefore no structural constraints. To avoid over-modeling, they first estimated the dimension of hidden states using cross-validation. To avoid over-fitting, they augmented their objective function with terms that favor sparsity. In the outer loop, a bootstrapping method was used to estimate the confidence intervals for all the parameters. Presence or absence of connection between the genes in the network depends on whether the confidence interval includes zero or not, which added up to be the structure of the network. The final result of their method was a connectivity matrix, but no dynamical model resulted since none of the estimated models agreed with the inferred connectivity matrix. Our work seeks to estimate parameters that agree completely with a given connectivity matrix.
Another approach is to separate the task of parameter estimation and model selection and perform them separately, as Gennemark and Wedelin (2007) did for S-system models of genetic networks, where parameter estimation was done under the constraint of currently estimated structure. This is useful also when connectivity is available from the literature (Iyer et al., 2001; Lee et al., 2002; Ren et al., 2000) in which case only parameters need to be estimated. Rangel et al. (2004a) recognized imposing constraints on parameters as a possible extension of their work and suggested it in the discussion, and we will be following that lead precisely in this report.

The purpose of this report is to develop state-space representations of genetic networks with known structures. The method we propose is to combine linear dynamical modeling with structural constraints to produce biologically realistic models that can predict the dynamic behaviors of genetic networks. The motivation for imposing structural constraints is as follows: Any genetic network has a structure specifying interactions between the genes represented as connections. Not all genes are connected, and in fact, every evidence points to genetic networks being sparse. The network structures can often be determined from experiments by biologists, or they can be roughly inferred by model selection. In linear dynamical models, the structure of the genetic network is mainly reflected in the elements of the system matrices. If there are no connections between the genes or connections between the gene and the external stimuli, their corresponding elements in the system matrices should be equal to zero. Without such constraints the models cannot take the structure of the network into account. Therefore, to ensure that estimated models agree with a known structure, constraints must be imposed on parameters. A popular method for the estimation of parameters in state-space models is the EM algorithm. However, conventional unconstrained EM algorithms cannot be applied to models with constraints, so modifications must be made. Incorporating network structure into the state-space model of genetic networks will lead to imposing constraints on the parameter space. Although constrained EM algorithms have been proposed in the engineering literature (Ahn and Oh, 2003; Welling and Weber, 2001), the proposed constrained EM algorithms require iterative numerical solutions to equations derived in the M-step of the EM algorithm. This iterative solution inside an already iterative method would make computation time intolerably long. While generalized EM algorithms like the one Wu et al. (1996) used, avoid iterative numerical solutions, they have slower convergence speed. Therefore, in this report, we present a new type of constrained EM algorithm that admits analytical and decoupled solution and thus preserves EM's speed while not resorting to numerical solutions or generalized EM. In the DBN community, this problem of structural constraint is called the known structure and partial observability for the learning of Bayesian networks (Murphy and Mian, 2002). Since structures of some genetic networks can be gleaned from the literature or discovered through ChIP-on-chip experiments (Lemmens et al., 2006), they should be taken into account whenever available (Zak et al., 2003). Application to synthetic data and real world SOS data show that our method significantly outperforms conventional EM and that structural constraints are important in the reverse engineering of genetic networks.

2 METHODS

2.1 Linear dynamical systems

We have adopted the linear state-space model as the underlying model for genetic networks, in particular the linear time-invariant (LTI) model. LTI is a linear state-space model where parameters do not change over time (Chen, 1999). A linear state-space model of a dynamical system can be written as

\[ x_{t+1} = Ax_t + Bu_t + w \]
\[ y_t = Cx_t + Du_t + v \]

(1)

where \( x_t \) is the state vector, \( y_t \) the output vector, \( u_t \) the input vector at time \( t \); \( w \) and \( v \) are independent noise terms assumed to be white Gaussian with zero mean and covariance \( Q \) and \( R \), respectively; matrix \( A \) is called the state transition matrix, \( B \) the input matrix, \( C \) the output matrix, and \( D \) the feed-forward matrix. Matrices \( A, B, C \) and \( D \), and covariance matrices \( Q \) and \( R \) together make up the parameters of the dynamical system. Of all the matrices, \( A \) is the most important, as its eigenvalues determine the stability of LTI. The system is stable if all eigenvalues are inside the unit circle in the complex plane and unstable otherwise. The states represent the biological forces that regulate gene expression. They describe the behaviors of gene transcription but are hidden. The outputs denote the gene expression levels and are measured by microarrays or green fluorescent proteins (GFPs): The expression level is determined by the states of the regulated gene. The inputs can be any external stimuli that influence gene regulation, such as drugs, proteins, RNAs or expression levels of connected genes.

The linear state-space model represented in Equation (1) is quite general and can represent more than simple exponential growth and decay, for it can represent higher order dynamics, which we will look at next.

2.2 Higher order dynamics

If we stick with one gene for one state, then the system represented in Equation (1) only will have first order dynamics associated with all the genes, which is exponential decay or growth, but since oscillation is widely observed in biology at least second order should be considered in models of genetic networks. We will give a simple derivation of how to add second order dynamics for the individual nodes of the networks using the principle of continuous to discrete conversion. This is similar to d’Alché-Buc’s method (d’Alché-Buc et al., 2005). Of course third or higher order dynamics can be similarly modeled, but care must be taken to avoid over-fitting.

Suppose we have a second order linear differential equation describing the dynamics of a node:

\[ \ddot{x} + \lambda_1 \dot{x} + \lambda_2 x = \sum_j w_j z_j \]

(2)

where \( x \) is the state of the node we are interested in, \( z_j \) is the expression level of node \( j \) and \( w_j \) its corresponding weight, and \( \lambda_1 \) and \( \lambda_2 \) parameters. Let

\[ x_1 = x, \quad x_2 = \dot{x} \]

Then we get

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\end{bmatrix}
= \begin{bmatrix}
\sum_j w_j z_j - \lambda_1 x_2 - \lambda_2 x_1 \\
-\lambda_2 & -\lambda_1 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\end{bmatrix}
+ \begin{bmatrix}
w_1 \\
\vdots \\
\end{bmatrix}
\]

(4)

If the steps are uniform, i.e. \( \Delta t = 1 \), then we can represent the derivatives as

\[
\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t}
\]

which becomes \( \Delta x = x(k+1) - x(k) \)

(5)
where \( k \) is the time step and the Equation (4) becomes
\[
\begin{bmatrix}
x_t(k + 1) \\
x_{t+1}(k + 1)
\end{bmatrix} = \begin{bmatrix}
x_t(k) + x_{t+1}(k) \\
\sum w_j z_j(k) - (1-\lambda_1 - \lambda_2) x_{t+1}(k) - \lambda_2 x_1(k)
\end{bmatrix}
\]
\[
= \begin{bmatrix}
1 & 1 \\
-\lambda_2 & 1 - \lambda_1
\end{bmatrix} \begin{bmatrix}
x_t(k) \\
x_{t+1}(k)
\end{bmatrix} + \begin{bmatrix}
w_1 \\
0 \ldots 1
\end{bmatrix} \begin{bmatrix}
z_t(k)
\end{bmatrix} \tag{6}
\]

The ones and zeros in Equation (6) are fixed except in \( 1 - \lambda_1 \) where the whole term is variable. An interesting observation is that all interactions and inputs are in the second order term \( x_{t+1} \).

We will apply this conversion to just one gene in the SOS network, lexA. But first we need to derive the constrained EM algorithm.

### 2.3 Expectation maximization (EM)

EM is a well known method in systems identification (Dempster et al., 1977; Ghahramani and Hinton, 1996; Gibson and Ninness, 2005; Shumway and Stoffer, 1982). EM is a Maximum-Likelihood (ML) estimator of unobserved states and unknown parameters and it operates in an iterative fashion. Each iteration consists of two steps: the E-step and the M-step. In the E-step, states are estimated using Kalman smoother with previous estimates of parameters as model parameters. In the M-step, parameters are estimated using the estimated states obtained in the E-step, and parameters are calculated as to maximize the likelihood. We will focus on our modification to the M-step where network structure constraints are taken into account. We will follow the notations in Gibson and Ninness (2005), while Kailath et al. (2000) is a good source on Kalman filter. Rewriting Equation (1) as
\[
\begin{bmatrix}
x_t \nxt \\
y_t 
\end{bmatrix} = \begin{bmatrix}
A & B \\
C & D
\end{bmatrix} \begin{bmatrix}
x_{t-1} \\
y_{t-1}
\end{bmatrix} + \begin{bmatrix}
w_t \\
\nu_t
\end{bmatrix}
\]
while adopting the following definition for the sake of convenience:
\[
x_t = \begin{bmatrix}
x_t \\
y_t
\end{bmatrix}, \Gamma = \begin{bmatrix}
A & B \\
C & D
\end{bmatrix}, \Pi = \begin{bmatrix}
Q & 0 \\
0 & R
\end{bmatrix}
\]

So Equation (7) becomes
\[
\xi_t = \Gamma \xi_t + \begin{bmatrix}
w_t \\
\nu_t
\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Pi \right)
\]

We also shall denote all the observations (or outputs) as \( Y_t \), all the inputs as \( U_t \), and all the states as \( X_t \). The E-step is explained in the Supplementary Materials since it is identical to that in conventional EM. Basically it calculates the conditional expectations for each time series
\[
E(y_t | X_{t-1}^N, U_{t-1}) = \tilde{z}_{t-1} \tilde{X}_{t-1} + P_{t-1}
\]

where \( \tilde{X}_{t-1} = E[X_t | Y_t, U_{t-1}], P_{t-1} = \text{var}[X_t | Y_t, U_{t-1}] \), and \( t_e \) is the number of time-points in the \( t \)th time series. \( Y_t \) and \( U_t \) are the observations and inputs for the time series. Based on the above, we can define:
\[
\Phi = \sum_{t=1}^{N} \sum_{t_e=1}^{t_e} E(y_t | x_t^N) (y_t^N, U_t^N) = \sum_{t=1}^{N} t_e
\]
\[
\Psi = \sum_{t=1}^{N} \sum_{t_e=1}^{t_e} E(y_t | z_t^N) (y_t^N, U_t^N) = \sum_{t=1}^{N} \sum_{t_e=1}^{t_e} E(z_t | z_t) (y_t^N, U_t^N)
\]

for all \( N \) time series needed for the M-step. Here the superscript denotes the \( t \)th time series.

The conventional EM would have the M-step as
\[
\mu \left( \tilde{x}_{t_{1:t_e}}, P_t \right) = P_{t_e}, \Gamma_{\text{new}} = \Psi \Sigma^{-1}, \Pi_{\text{new}} = \Phi - \Psi \Sigma^{-1} \Psi^T \tag{10}
\]

where \( \mu \) is the estimated mean of \( x_t \) and \( P_t \) the estimated variance. We note here that the last two equations in (10) are decoupled and everything on the right hand side can be computed from the E-step results; thus the M-step in conventional EM has an analytic solution and is very fast. This feature of conventional EM is one reason for its popularity, but it may be lost if constraints are imposed on parameters because structural constraints force some parameters to be zeros while leaving others free to change. In that case, Equation (10) is no longer valid due to structural constraints, and numerical solution may be required for maximization. Having an iterative solver within an already iterative method will significantly increase computation time. Generalized EM, another solution that can permit constraints on parameters however, is known to have slower convergence. Since parameter estimation could become the inner loop of a bigger model selection algorithm, we want to strive for decoupled and analytic solutions. Fortunately, with a mild assumption on the type of noise, that it has diagonal variance, we are able to obtain an analytic solution.

### 2.4 Structural systems identification

Given a network whose structure is known, for example if a pair of genes is represented by two states and they have no interaction, then the corresponding entry in matrix \( A \) should be zero. Similarly, if an input has no influence on a gene that is represented as a state, then the corresponding entry in \( B \) should be zero. The same goes for entries in \( C \) that describe how measurements depend on the states. Matrix \( D \) is usually all zeros because genes do not impact other genes’ expression level instantaneously.

Take for example the SOS network in Figure 4. The sole input is the gene recA, so input vector \( a_t \) is a scalar, and matrix \( B \) is a vector whose entries are all zeros except for the first element. Among the rest of the genes, only lexA interacts with other genes so matrix \( A \) only has the first column and the diagonal entries as non-zero for a first order system. The matrix \( A \)'s diagonal entries are non-zero because, in general, genes impact their own expression levels. Since we measure all the genes’ expression levels directly, we set \( C \) to be an identity matrix. \( D \) is a zero matrix as explained above. So putting all this together, we have parameters \( A, B, C \) and \( D \) initially determined as follows:
\[
A = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_{21} & 0 & 0 & 0 \\
0 & 0 & 0 & a_{31} & 0 & 0 \\
0 & 0 & 0 & 0 & a_{41} & 0 \\
0 & 0 & 0 & 0 & 0 & a_{51} \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
\[
B = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]
\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]
\[
D = 0
\]

But in initial estimations, we discovered that first order dynamics does not adequately describe the time series, so we increased the order of the lexA gene to two and that proved successful. Accordingly, the parameters \( A, B, C \) and \( D \) are changed to
\[
A = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{22} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_{33} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_{44} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & a_{55} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a_{66} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & a_{77}
\end{bmatrix}
\]
\[
B = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]
\[
C = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
\[
D = 0
\]
Here we expanded gene lexA to have second order dynamics, resulting in two scalar states, $x_1$ and $x_2$, where $x_2$ is the derivative of $x_1$ discretized, while $x_1$ represents lexA’s expression level. Matrix $B$ is changed because all interactions are on the second order term, and $C$ is changed to reflect the fact that we do not have measurement for $x_2$ (column 2 is all zeroes). We also assume $\Pi$ to be diagonal since we have no reason to believe that noise in each state or measurement is correlated. This also is a standard assumption unless there is specific evidence that contravenes the assumption. Making $\Pi$ diagonal also makes the analytic form of $M$-step possible.

Incorporating structural constraints results in an $M$-step that is more complicated than the $M$-step in the conventional EM algorithms:

\[
[\Gamma_{\text{new}} \Sigma - \Psi] \odot M = 0
\]

(11)

where $I$ is an identity matrix of appropriate size and $M$ is a constraint matrix of $\Gamma$ so that if an entry of $\Gamma$ is constrained, the corresponding entry of $M$ is 0, and 1 otherwise. The notation $\odot$ represents element-wise product, also known as the Hadamard product. Equations (11) and (12) are quite general, so the formulas admit non-zero constrained values.

Equation (12) looks more complicated, but the calculation is actually explicit as long as we have $\Gamma_{\text{new}}$ from Equation (11). Equation (11) can be solved row-wise by the following procedure:

Explicit solution for $\Gamma$ in $[\Gamma \Sigma - \Psi] \odot M = 0$ can be obtained by:

1. for each row of $\Gamma$, $\Gamma_i$, suppose $r_i$ = \{indices of constrained elements of $\Gamma_i$\};
2. delete all elements of $\Gamma_i$ and $\Psi_i$, the $i$th row of $\Psi$, whose indices are in $r_i$;
3. delete all rows and columns of $\Sigma$ whose indices are in $r_i$;
4. solve $[\Psi]_{\text{mod}} - [\Sigma]_{\text{mod}}[\Gamma]_{\text{mod}} = 0$ where the $[\_]_{\text{mod}}$ notation denotes respective vectors and matrices after deletion in step 2 and 3.

Therefore the procedure above and Equation (12) plus the first two equations in Equation (10) constitute the modified $M$-step.

2.5 Data source

First, the synthetic data was generated by a system that had four states and four outputs and the parameters were as follows:

\[
A = \begin{bmatrix}
0.8 & 0.0 & 0.8 & 0 \\
0.0 & 0.8 & 0.0 & 0.8 \\
0.0 & 0.0 & 0.8 & 0.0 \\
0.8 & 0.0 & 0.0 & 0.8
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
-1 \\
-1 \\
-1 \\
-1
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
D = 0
\]

We generated 200 time points for cross-validation. Since $C$ is fixed in estimation as an identity matrix, no equivalent system exists by similarity transformation, and this system is identifiable.

Second, to validate our method with real-world data, we chose the SOS DNA repair network of the Escherichia coli with eight essential genes. The SOS network is a highly conserved system and is a well studied network (Camas et al., 2006; Friedman et al., 2005). It consists of 30 genes, the master regulator being the lexA gene. Gene lexA inhibits all the rest of the SOS network’s gene under normal condition, and when DNA damage is sensed, the normally suppressed genes become active. A diagram of SOS network with eight essential genes is shown in Figure 4.

The experimental data for the SOS system can be downloaded from Uri Alon’s homepage. Ronen et al. (2002) used green fluorescent proteins (GFP) to track eight genes of the SOS network as they react to different irradiation levels, 5Jm$^{-2}$ and 20Jm$^{-2}$; each level has two samples and each sample has 50 evenly spaced time points. They monitored eight genes: uvrD, lexA, umuD, recA, uvrA, ruvA and polB. They performed extensive data preprocessing on the raw data using hybrid Gaussian median filter and polynomial fit for smoothing. They also assumed that the rate of accumulation of GFP was proportional to transcript production. We shall make the same assumption.

To test our method on more complex models, we decided to use a biologically inspired artificial system by Zak et al. (2003). They used stochastic simulation to simulate gene expressions and protein interactions. The data had 550 time points, but very few time-course data currently available are that long. So we sampled one time point out of every five consecutive time points to obtain a time series of 110 time points. A complete diagram of the artificial system is shown in Figure 5, and the parameters are constrained as follows:

\[
A = \begin{bmatrix}
a_{11} & 0 & 0 & a_{14} & 0 & 0 & 0 & 0 & 0 \\
a_{21} & a_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{31} & 0 & a_{33} & 0 & a_{35} & 0 & 0 & 0 & 0 \\
0 & a_{41} & a_{44} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_{53} & 0 & a_{55} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & a_{65} & a_{66} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a_{76} & a_{77} & 0 & 0 \\
0 & 0 & 0 & 0 & a_{85} & a_{88} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & a_{89} & a_{99}
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
-1 \\
-1 \\
-1 \\
-1
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
D = 0
\]

3 RESULTS

To examine the consistency of the constrained EM approach and to test its biological applicability, we applied our new method to two sets of synthetic data and the SOS DNA repair network data. We first examined the distributions of standardized residuals (or errors) of the Kalman filter for the synthetic data and found that they largely resemble Gaussian distributions (these results are presented in Supplementary Materials). For the synthetic data sets, we compared the predictive power and estimation precision of our constrained EM and the unconstrained EM through prediction errors and confidence intervals. For the SOS data, two replications were not enough for bootstrapping so no confidence interval could be derived.

To compare the predictive power of identified models from conventional EM and models from our constrained EM, we used cross-validation. For our synthetic data, we generated a
sample of 200 time points, of which the first 100 were used for parameter estimation, and the rest for prediction. Of Zak’s data, we selected 110 time points, out of which the first 80 time points were used for estimation and the remaining 30 for validation. The error in prediction is defined to be the difference between the measured gene expression levels and the predicted gene expression levels by the estimated model. Using constrained EM, the error in prediction for four gene expression data in the artificial network from $t = 101$ to $t = 200$ were calculated (Fig. 1). In Supplementary Figure 3, for comparison, we plotted the error in prediction by conventional EM. From the plots, we can see that conventional EM starts off with small errors, and sometimes it produces smaller errors initially than our method. But very quickly it strays into wrong directions with larger and larger errors, which makes models estimated from conventional EM having little predictive power. This is even more striking with Zak’s data. The constrained EM algorithm yields models with prediction error at worst around 100%, with the output of the worst error plotted in the top panel of Figure 3, while unconstrained EM generates models whose prediction error grows without bound as seen in the bottom panels of Figure 3. In fact, we were forced to cut the plot into three panels so that later values would not obscure earlier ones. This demonstrates that conventional EM, which does not take into account the structure, tends to over-fit.

To further test our method we considered the variation in the estimated values. Since many parameter values could fit data equally well, confidence intervals are usually preferred over a single estimation. Using bootstrapping, we were able to estimate 95% confidence intervals for all free parameters and the eigenvalues of the estimated system [same as the eigenvalues of matrix $A$ in Equation (1)]. The idea is that tighter intervals are better than wide intervals and that the estimation methods need to get eigenvalues roughly right, since they are invariant to similarity transformation and they determine important dynamical properties. We found that for almost all free parameters, our constrained EM produced much tighter bounds than those produced by unconstrained EM (see Supplementary Material Table 1 for selected parameters), and for all eigenvalues, our method uniformly produced better bounds (presented in Supplementary Table 2). As we can see from Supplementary Table 2, estimated eigenvalues for our simulated model all include the true eigenvalues 0.8, but our method has much tighter bounds. For Zak’s model, unconstrained EM resulted in one eigenvalue having wide interval while constrained EM all have tight bounds. That wide interval,
much of it outside the unit circle, could be a sign of a misestimated eigenvalue, since it could account for the unbounded prediction error we observed. A misestimated eigenvalue is a serious concern because the trajectory of an LTI system largely depends on its eigenvalues.

To examine the biological applicability of our method and to evaluate its performance with real world data, we applied our constrained EM to the SOS DNA repair network data. Since there are only 50 time points available, we used the last 10 time points for validation purpose. The error in prediction of the seven-gene expression data in the SOS DNA repair network by the constrained and the conventional EM are plotted in Figure 2 and in Figure 4 of Supplementary Materials. Two features of the plots can be observed. First, the errors oscillate within a certain range rather than blow up in one direction. This suggests that with fewer time points conventional EM performs better than with more data, a classic sign of over-fitting. Second, the constrained EM’s estimated model has much smaller deviation from measurement than conventional EM’s estimated model, once again demonstrating constrained EM’s merits. The figures show that the constrained EM better approximates the true model of the SOS DNA repair network and generate models with superior predictive power.

4 DISCUSSION
It is increasingly recognized that dynamics of genetic networks can impact phenotypes, and the study of dynamics can provide new insights into diseases and potential treatments for the diseases (Strohman, 2002). But before we can analyze genetic networks and propose possible treatments, we need a quantitative model that can predict, with reasonable accuracy, the dynamical behaviors of the genetic network. In this report, we proposed a new method that can learn such a model, a linear state-space system, and tested on both synthetic and real experimental data.

Researchers so far have used the linear state-space model primarily in two ways. Either they are used in black-box dynamical modeling or in inferring a genetic network’s structure. Wu et al. (2004) is representative of the first approach, where the internals of the model is not important but only the dynamical behaviors is, where the number of states is a
parameter depending on the data, and where the states have no biological interpretation. This black-box approach is perfectly valid when we have little information regarding the mechanistic details. However, sometimes we have structural information for some genetic networks, either through existing knowledge in the literature or ChIP-on-chip experiments, in which case parameter estimation should take the known structure into account to get a better model. A better model is in the sense that the estimated model in the end does not contradict known biological facts represented by the structure and possess better predictive power. Another use of linear state-space modeling is to infer a genetic network’s structure. Using the linear model for network inference is especially appealing because the structure and the parameters have a simple relationship: there is a straightforward mapping between parameters and edges in the network. A naive approach would be to estimate a model with all parameters free to be estimated and to consider those parameters whose values are below a threshold as really being zeros and thereby signify no interaction. We have seen that conventional EM tends to over-fit and produce a model that has limited predictive power. One approach to alleviate over-fitting is to enforce sparsity on the parameters (d’Alché-Buc et al., 2005). Another approach is to separate parameter estimation and structural inference, to incorporate structural constraints into parameter estimation, as in Gennemark and Wedlin (2007). Our method can be seen as the parameter estimation part of the overall system identification, which also infers structure.

In order to incorporate structural constraints into an identification of genetic networks in this report, we presented a framework where certain parameters are fixed while others remain variable. Imposing constraints on parameters lead to a set of non-linear equations to be solved in the M-step. In order to have fast convergence, we intentionally avoided generalized EM (which is slower than our method) or using iterative solutions to the set of non-linear equations in the M-step (which can also be very slow). Instead, with only mild assumptions about noise, we obtained a closed-form, decoupled, explicit solution to the equations arising from the maximization of likelihood in the M-step.

To evaluate the performance of our new method, we applied it to two synthetic data sets and a real world SOS DNA repair network data set. From the results, we can see positive features of incorporating structural constraints in general and constrained EM in particular. First, by incorporating known connectivity between genes, we have better biological realism along with a biological interpretation for the identified model. Second, a state-space model with structural constraints exploits the sparse nature of genetic networks to reduce the number of parameters that need to be estimated. We know that reverse engineering of genetic networks is a grossly underdetermined task, and that we have too few data for a fully connected network. However, the problem can be alleviated if we can effectively incorporate more information by imposing structural constraints. Increasingly, connectivity information is becoming available for more genetic networks (Iyer et al., 2001; Lee et al., 2002; Ren et al., 2000), and it becomes necessary for us to use this information. Third, by using analytic, explicit solution to the maximization of likelihood, we have fast computation, so that our method can be easily incorporated into larger structural inference algorithms. The last and the most important feature of our work is that by incorporating structural constraints into an identified model we identify models that are not prone to over-fitting, that have better predicative power than unconstrained models, and, therefore, are closer to the true model. Cross-validation using both the synthetic data and the SOS data demonstrated this point. With better predictive power, the identified model can then be used for the analysis of dynamical properties or for the design of control strategies for the genetic network under study. There is considerable work that remains. On the modeling front, our method does not take into account the signs of parameters, which represent whether regulation is activation or inhibition. This is crucial information that should be incorporated as another set of constraints. A problem of a more theoretical nature is that we have as yet no principled way to determine the order of each gene, as too high an order can result in over-fitting and too low an order leaves too much unmodeled dynamics. Finally, as LDS is only a linear approximation of gene regulation systems, care must be exercised in extrapolating results presented here to real world expression data. Much more work on diagnostics and model validation remains as well.

Conflict of Interest: none declared.

REFERENCES