Genetic network identification using convex programming

A. Agung Julius[1,*], Michael Zavlanos[1,*], Stephen Boyd[2,†], and George J Pappas[1,*]

1. Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia PA 19104, USA.
2. Department of Electrical Engineering, Stanford University, Stanford CA 94305, USA.

*email: {agung,zavlanos,pappasg}@seas.upenn.edu, †email: boyd@stanford.edu

Introduction

Genes in living cells regulate various cellular biochemical processes through genetic regulatory networks. In such a network, the genes produce proteins that act as transcription factors for other genes or themselves. The use of RNA microarrays has made it possible to have an expression profile for a large number of genes when exposed to different conditions. One of the most important problems in systems biology is to use these data to identify the interaction pattern between genes in a regulatory network, especially in a large scale network. In the literature, this is sometimes called *reverse engineering* the genetic network (see the survey paper [1]). Genetic network identification has important potential applications, for example in drugs discovery where a systems wide understanding of the regulatory network is crucial for identifying the targeted pathways.

In this paper we propose a method for identifying genetic regulatory networks using *genetic perturbation* data. In a genetic perturbation experiment, small perturbations are applied to a genetic network in an equilibrium state and the resulting changes in expression activity are measured. We aim at identifying the smallest model, corresponding to the sparsest network, that explains the data, while conforming to known *a priori* structural information about the network, if any. A priori biological knowledge is typically *qualitative*, encoding whether one gene affects another gene or not, or whether the effect is positive or negative. We solve the combinatorially hard problem of finding the sparsest model using a technique from convex optimization [2], and demonstrate that our method performs better than other existing methods.

Approach

A genetic regulatory network consisting of n genes in a genetic perturbation experiment can be modeled as an $n-$dimensional dynamical systems ([3, 5]).

$$\frac{d\hat{x}}{dt} = f(\hat{x}, u),\ \hat{x} \in \mathbb{R}^n,\ u \in \mathbb{R}^p, \tag{1}$$

where $\hat{x}_i$ denotes the transcription activity (typically measured as mRNA concentration) of gene i in the network, and u_i is the so called transcription perturbation. The dynamics close to a given equilibrium x_{eq} can be approximated by the set of linear differential equations,

$$\frac{dx}{dt} = Ax + Bu, \tag{2}$$

where $x := \hat{x} - x_{eq}$ ([4]). The matrix $A \in \mathbb{R}^{n\times n}$ encodes pairwise interactions between the individual genes in the network at the given equilibrium (phenotypical state), while matrix

$B \in \mathbb{R}^{n\times p}$ indicates which genes are affected by the transcriptional perturbations. Given that the system is stable around the equilibrium $x = 0$, if u is small enough, the system will move to a new equilibrium x, for which $Ax + Bu = 0$. Let $U = [U_1 \ \dots \ U_m] \in \mathbb{R}^{p\times m}$ denote the stack matrix of the transcription perturbations for different m experiments and $X = [X_1 \ \dots \ X_m] \in \mathbb{R}^{n\times m}$ denote the stack matrix of the corresponding steady state mRNA concentrations.

Assuming that the measurements X and U are corrupted by noise, the equilibrium condition becomes $AX + BU = \eta$, where η is the identification error. The goal of our method is to find unknown matrix A, which models genetic network interactions and minimizes η with respect to some metric. The error criterion that we choose is the weighted total squared error,

$$J(A) = \sum_{j=1\dots m} \eta_j^T R_j \eta_j, \text{ where } \eta_j := AX_j + BU_j. \tag{3}$$

The weight matrix R_j is the inverse of the covariance of the measurement error in the $j-$th experiment. This means we penalize the identification error more when the measured data is more reliable (less noisy). The *a priori* structural constraint that we impose on the network can be encoded as $A_{ij}\square 0$, where $\square \in \{<, =, >\}$. The set of all $n \times n$ matrices that satisfy the *a priori* constraint is convex, and we denote this set by $\mathcal{S}$. The best model that satisfies the *a priori* constraint while minimizing the error criterion can be found by solving the following convex optimization problem.

$$\text{minimize } J(A), \text{ subject to } A \in \mathcal{S}. \tag{4}$$

We denote the obtained minimum error level as the *baseline error level* (E_{bs}). Subsequently, we find the sparsest model by solving the optimization problem

$$\text{minimize } \|A\|_0, \text{ subject to } A \in \mathcal{S},\ J(A) \le \beta E_{\text{bs}}, \tag{5}$$

where $\|A\|_0$ is the number of nonzero entries in A, and β is a parameter that controls the tradeoff between model minimality and model accuracy.

Problem (5) is combinatorially hard and not convex. To solve this problem efficiently, we relax it as a recursive convex ℓ_1 optimization problem. That is, we want to find the sequence of matrices $A^{(k)}$, $k = 0, 1, \dots$ from the following convex optimization problem.

$$\begin{array}{ll} \text{minimize} & \sum_{i,j} W_{ij}^{(k)} \left|A_{ij}^{(k)}\right| \\ \text{subject to} & A \in \mathcal{S},\ J(A) \le \beta E_{\text{bs}}, \text{ where} \\ W_{i,j}^{(0)} = 1, & W_{i,j}^{(k+1)} = \frac{1}{1+1000\cdot\left|A_{ij}^{(k)}\right|}. \end{array} \tag{6}$$

This ℓ_1 relaxation technique has been applied successfully in various fields where sparsity optimization is needed such as, portfolio optimization in finance, controller design in engineering, and electric power network design.

Results

We implement our method on MATLAB using the cvx toolbox[1], running on an Intel Xeon

[1]More efficient implementation of the algorithm could possibly handle larger problems, but might require custom made software.

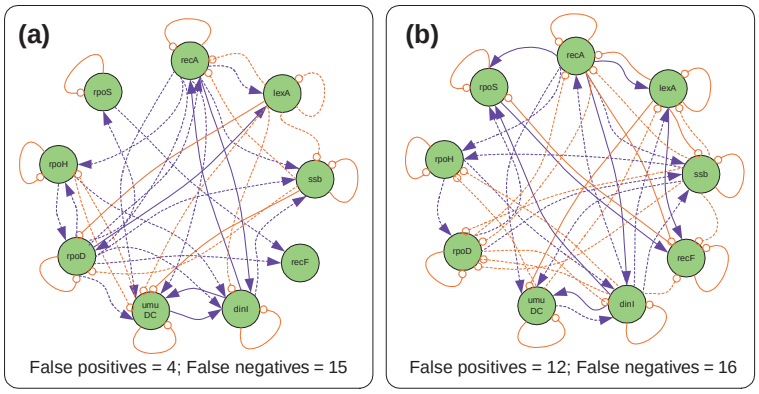

Figure 1: Identified models of the *Escherichia coli* SOS network. **(a)** The result of our method. **(b)** The network identified in [3].

2.8GHz processor with 4GB RAM. We apply our method on the following data sets.

The segmentation polarity network in *Drosophila melanogaster*. We obtain a data set from an *in silico* model provided by [5]. The original network consists of 5 genes. Our method takes 6 seconds to run and identifies a smaller model than that in [5] with higher accuracy (fewer false positives).

The SOS pathway in *Escherichia coli*. We obtain experimental data set from [3]. The data set consists of 9 genes, and the performance comparison between our method and that of [3] is shown in Figure 1.

A larger artificial network. We construct an artificial random network of 20 genes and generate a noisy data set from it. For a noise level of 10%, our method takes about 9 minutes to run and produces a result with predictive positive values and sensitivity of higher than 90%. This is better than the benchmark results from other methods reported in [1].

References

[1] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3(10.1038/msb4100120), 2007.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301:102–105, 2003.

[4] E. Sontag, A. Kiyatkin, and B. N. Kholodenko. Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data. *Bioinformatics*, 20(12):1877–1886, 2004.

[5] J. Tegner, M. K. S. Yeung, J. Hasty, and J. J. Collins. Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. *Proc. of the National Academy of Science*, 100(10):5944–5949, 2003.