Gene Regulatory Network Inference from Perturbed Time-Series Expression Data via Ordered Dynamical Expansion of Non-Steady State Actors

Mahdi Zamanighomi, Mostafa Zamanian, Michael Kimber, and Zhengdao Wang

Abstract—The reconstruction of gene regulatory networks from gene expression data has been the subject of intense research activity. A variety of models and methods have been developed to address different aspects of this important problem. However, these techniques are narrowly focused on particular biological and experimental platforms, and require experimental data that are typically unavailable and difficult to ascertain. The more recent availability of higher-throughput sequencing platforms, combined with more precise modes of genetic perturbation, presents an opportunity to formulate more robust and comprehensive approaches to gene network inference. Here, we propose a step-wise framework for identifying gene-gene regulatory interactions that expand from a known point of genetic or chemical perturbation using time series gene expression data. This novel approach sequentially identifies non-steady state genes post-perturbation and incorporates them into a growing series of low-complexity optimization problems. The governing ordinary differential equations of this model are rooted in the biophysics of stochastic molecular events that underlie gene regulation, delineating roles for both protein and RNA-mediated gene regulation. We show the successful application of our core algorithms for network inference using simulated and real datasets.

Index Terms—Gene regulatory networks, network inference, time-series gene expression, perturbation, RNA-Seq, sparsity, constraint optimization

1 Introduction

The elucidation of gene regulatory networks is fundamental to understanding the dynamic functions of genes in biochemical, cellular and physiological contexts. The architectures of networks comprised of small numbers of genes are generally deciphered using classical experimental techniques, where biophysical data describing the interactions of genes and their products can lead to useful models and well-characterized systems. While this validated experimental tract continues to provide valuable biological insight, it is ultimately laborious and costly, and often demands strategies uniquely tailored to individual biological systems and problems. Furthermore, the models that result from these efforts tend to be limited to a very modest subset of genes, typically suffer from a lack of temporal resolution, and focus narrowly on very particular modes of interaction.

To complement these established approaches, there is a great impetus to develop more efficient and uniformly applicable *in silico* methods for gene network inference and discovery [1], [2], [3], [4], [5], [6], [7]. Of particular interest is the goal of gene network inference using perturbed gene expression data [8], [9], [10], [11], [12], [13], [14], [15], [16], whereby gene expression levels are measured under the influence of either genetic or chemical perturbations of the system. Previous attempts at network reconstruction via perturbation tend to be limited to the analysis of steady-state gene expression. The growing ubiquity of next-generation sequencing technologies presents a powerful high-throughput substrate for capturing the dynamic and non steady-state aspects of gene expression.

In this work, we seek to develop a robust framework for network inference that relies on temporal gene expression data coupled to genetic or chemical perturbation. In a departure from previous attempts, our formulation does not require *a priori* knowledge beyond the set of temporal gene expression measurements, acknowledges the non-steady state and dynamic nature of gene expression, incorporates both RNA and protein-mediated regulation, sequentially absorbs a growing number of genes into the regulatory network immediate to perturbation, aims for sparsity in network topology, and reduces an otherwise complex optimization problem into a convex form that can be solved efficiently. This method is best suited for small network inference local to a point of perturbation, within a larger network in approximate steady state.

- *M. Zamanighomi is with the Department of Statistics, Stanford University, Stanford, CA 94305. E-mail: mzamani@stanford.edu.*
- *M. Zamanian is with the Department of Molecular Biosciences, Northwestern University, Evanston, IL 60208, and the Department of Biomedical Sciences, Iowa State University, Ames, IA 50010. E-mail: zamanian@northwestern.edu.*
- *M. Kimber with the Department of Biomedical Sciences, Iowa State University, Ames, IA 50010. E-mail: michaelk@iastate.edu.*
- *Z. Wang is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50010. E-mail: zhengdao@iastate.edu.*

Manuscript received 18 Feb. 2015; revised 14 Sept. 2015; accepted 23 Nov. 2015. Date of publication 17 Dec. 2015; date of current version 6 Aug. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2015.2509992

1545-5963 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Notation. Throughout this paper $\{d, i, j, k, l\}$ count integer numbers. Column vectors and matrices are indicated by bold lower-case and upper-case letters, respectively. We use $\mathbf{1}$ to show a vector with all entries 1 and $\mathbf{0}$ a vector with all entries 0. The set of real numbers is denoted as $\mathbb{R}$ and positive real numbers $\mathbb{R}^+$. The indicator function $\mathbb{I}_{\mathbb{R}^+}\{x\}$ has the value one when $x \in \mathbb{R}^+$, otherwise zero. The operator $\text{sign}(\mathbf{x})$ replaces each entry of $\mathbf{x}$ with its sign function value. We use $(\mathbf{X})^T$ to denote transpose of $\mathbf{X}$, $dx(t)/dt$ and $x'(t)$ the first derivative of $x(t)$ with respect to time t, $\|\mathbf{x}\|_1$ the 1-norm of vector $\mathbf{x}$, $\|\mathbf{x}\|_2$ the 2-norm of vector $\mathbf{x}$, and $\|\mathbf{X}\|$ the largest singular value of matrix $\mathbf{X}$. We explicitly state a function of time in the form $\mathbf{x}(t)$. This is to be distinguished from vectors of the form $\mathbf{x}(i)$, where i is a positive integer representing the ith entry of the vector $\mathbf{x}$.

2 System Model

2.1 Gene Expression Datasets and Perturbation

Let $x_i(t)$ and $y_i(t)$ denote the RNA-level and protein-level expression of gene i at time t, respectively. In practical cases, with expression data originating from microarray or RNA-Seq experiments, the total number of genes in the system is exceedingly larger than the total number of samples in the time series.

The paper is concerned with datasets with known points of perturbation. In this experimental scheme, a gene x_i^p is specifically targeted for perturbation via either gene suppression or gene over-expression. Perturbation is triggered at a known time point after a series of presumably steady state measurements. Without loss of generality, it is assumed that the starting point of perturbation occurs at t_1 and prior measurements are approximately steady state. Datasets from experiments that conform to this scheme are in the following form, where $x_i^p(t_1)$ represents the point of perturbation and L denotes the total number of samples post-perturbation

$$\mathbf{X}^p := \begin{pmatrix} \dots & x_1(t_0) & x_1(t_1) & \dots & x_1(t_L) \\ \ddots & \vdots & \vdots & \ddots & \vdots \\ \dots & x_i(t_0) & x_i^p(t_1) & \dots & x_i^p(t_L) \\ \ddots & \vdots & \vdots & \ddots & \vdots \\ \dots & x_m(t_0) & x_m(t_1) & \dots & x_m(t_L) \end{pmatrix}.$$

2.2 Conceptual Description of Inference Approach

We consider a non-perturbed system as one with genes in steady state, i.e., where $dx_i(t)/dt$ and $dy_i(t)/dt$ are approximately zero. After a series of steady state expression measurements, a protein-encoding gene in this system is perturbed to bring about a dramatic change in its expression level, i.e., where $|dx_i^p(t)/dt| \gg 0$, followed by a series of post-perturbation measurements. The discrete set of expression measurements, with appropriate temporal resolution, can be used to produce continuous gene trajectory curves.

For a short period of time post-perturbation, the perturbed gene falls out of steady state while all other genes remain effectively in steady state. The induced change in RNA expression, Δx_i^p, is coupled to a delayed change in protein expression, Δy_i^p. This shift in protein availability leads, through the immediate regulatory network of the perturbed protein, to changes in the expression levels of other genes.

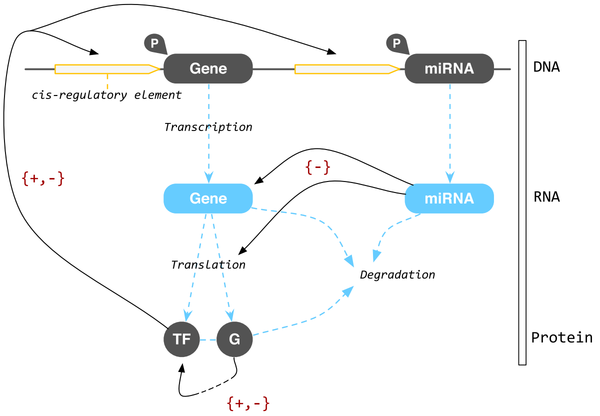


Fig. 1. Gene regulatory circuit. ‘Gene’ represents protein-encoding genes and ‘miRNA’ represents miRNA-encoding genes. Protein-encoding genes can give rise to transcription factors (‘TF’) that directly exert influence on the cis regions of other genes, as well as non-TF proteins (‘G’) that can indirectly act through TFs and various biochemical cascades. These protein regulators ultimately affect the equilibrium probability of RNA polymerase (‘P’) being bound to a promoter of interest. Additionally, miRNAs can directly repress expression via targetted RNA degradation or translational repression. All proteins and RNAs in this system undergo varying rates of chemical degradation.

Consider the set of all genes that are affected by Δy_i^p at time t. We divide this set into protein and miRNA-encoding subsets. The set of all indices that correspond to protein-encoding genes is shown as $G(t)$, and $M(t)$ is set of all indices that correspond to miRNA-encoding genes. We define the collection of RNA expression data for these subsets as $\mathcal{X}_{G(t)} := \{x_i(t) | i \in G(t)\}$ and $\mathcal{X}_{M(t)} := \{x_i(t) | i \in M(t)\}$, respectively. We further define the collection of protein expression levels for subset G as $\mathcal{Y}_{G(t)} := \{y_i(t) | i \in G(t)\}$.

In principle, we can identify genes that fall out of steady state in an ordered manner with gene trajectory analysis. The growing set of non-steady state actors in the system, both members of $G(t)$ and $M(t)$, can then be sequentially incorporated into a growing network of interactions to be modeled.

2.3 Governing Regulatory Equations

Gene and protein expression dynamics are often modeled in the form of ordinary differential equations (ODE) [17], [18], [19], with gene-specific rate constants for molecular synthesis and degradation and gene-specific functions accounting for the regulatory effects of proteins. We introduce miRNA-mediated gene regulation into this model and establish functions for both protein and RNA regulatory interactions that complement our overall approach to network inference. The architecture of the gene regulatory circuit under consideration is depicted in Fig. 1.

This circuit can be represented in the following form:

$$\frac{dx_i(t)}{dt} = \tau_i f_i(\mathcal{Y}_{G(t)}) - \big(\lambda_i^{RNA} + g_i(\mathcal{X}_{M(t)})\big)x_i(t), \qquad (1)$$

$$\frac{dy_i(t)}{dt} = \big(r_i - h_i(\mathcal{X}_{M(t)})\big)x_i(t) - \lambda_i^{Prot} y_i(t), \qquad (2)$$

where τ_i is the rate of transcription when RNA polymerase (RNAP) is bound, $f_i(\mathcal{Y}_{G(t)})$ is the probability of RNAP binding, λ_i^{RNA} is the rate of basal RNA degradation, $g_i(\mathcal{X}_{M(t)})$

incorporates the effect of miRNA-mediated RNA degradation, r_i is the rate of translation, $h_i(\mathcal{X}_{M(t)})$ accounts for the effect of miRNA-mediated translational inhibition, and λ_i^{Prot} is the rate of protein degradation. It follows from the biological definitions of the system that parameters τ_i, λ_i^{RNA}, r_i, and λ_i^{Prot} are to be positive and $h_i(\mathcal{X}_{M(t)}) \leq r_i$.

2.4 Protein-Mediated Regulation

For each gene, i, we employ an existing statistical thermodynamic framework [20], [21] to model the equilibrium probability of RNAP binding to a gene of interest as a function of protein regulators, $f_i(\mathcal{Y}_{G(t)})$. We extend a previous derivation of multiple protein regulators operating on a single gene [22] and explicitly show that the general form can be expressed as a function of non-steady state genes, $G(t)$ (Appendix A). Although steady state regulators play an active role in gene regulation, we can effectively restrict our binding probability function to the activities of perturbed regulators. This function is shown below

$$f_i(\mathcal{Y}_{G(t)}) = \frac{a_{i0} + \sum_{j=1}^{N(t)} a_{ij} \prod_{k \in S_{ij}(t)} y_k(t)}{1 + \sum_{j=1}^{N(t)} b_{ij} \prod_{k \in S_{ij}(t)} y_k(t)}, \tag{3}$$

where $S_{ij}(t)$, $0 \leq j \leq N(t)$, is the list of all possible protein products of genes within set $G(t)$ that interact to form regulatory complexes and $N(t)$ is the cardinality of set $S_{ij}(t)$. For instance when $G(t) = \{1, 2\}$, there are $N(t) + 1 = 4$ complexes as the empty set $S_{i0} = \{\emptyset\}$, $S_{i1} = \{1\}$, $S_{i2} = \{2\}$, and $S_{i3} = \{1, 2\}$. To reduce the complexity of this model, we restrict $S_{ij}(t)$ to all terms up to the second-order, accounting for the interactions of no more than two proteins bound together. In this arrangement, a complex represents either the products of a single gene or the interaction of the products of any two genes that can form a regulatory agent. However, any number of complexes can additively combine to regulate single genes. The numbering of complexes is an arbitrary labeling of genes and gene-pairs in the system. As shown in Appendix A after equation (19), the coefficients $0 \leq a_{ij} \leq b_{ij}$ depend on the binding energies of regulator complexes that act on a promoter region, and a_{i0} and b_{i0} correspond to the case where no regulators are bound to the promoter region ($\prod_{k \in S_{i0}(t)} y_k(t) := 1$). It is assumed all coefficients are normalized so that $b_{i0} = 1$.

2.5 miRNA-Mediated Regulation

To account for the effects of miRNA regulation on each gene, we draw on previous mass-law (linear) models [23], [24] that acknowledge two primary routes of inhibitory regulation: (i) cleavage or degradation of target transcript and (ii) translational repression. These are represented by functions $g_i(\mathcal{X}_{M(t)})$ and $h_i(\mathcal{X}_{M(t)})$, respectively. The former is a modifier of the RNA degradation rate constant, λ_i^{RNA}, while the latter detracts from RNA available to the translational machinery without affecting RNA concentration as assayed. These functions are shown below

$$g_i(\mathcal{X}_{M(t)}) = \sum_{j \in \mathcal{X}_{M(t)}} \lambda_{ij}^{RNA} x_j(t), \tag{4}$$

$$h_i(\mathcal{X}_{M(t)}) = \sum_{j \in \mathcal{X}_{M(t)}} \lambda_{ij}^{Prot} x_j(t), \tag{5}$$

where both λ_{ij}^{RNA} and λ_{ij}^{Prot} are greater than or equal to zero.

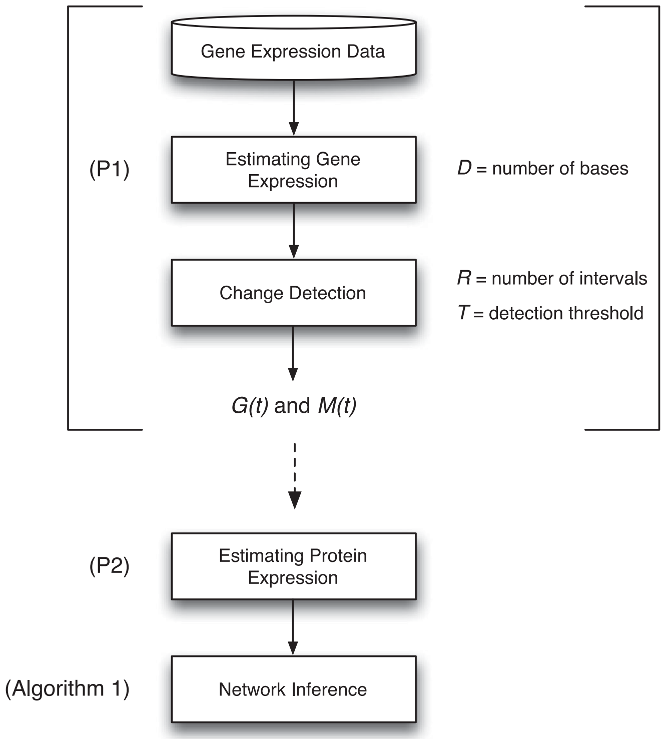


Fig. 2. Overview of gene inference pipeline, beginning with a normalized gene expression dataset. The first stage involves the estimation of all gene trajectories as noise-free and continuous curves (P1), followed by segmentation into equally-spaced intervals for detection of significant changes in expression. The time-dependent expansion of $G(t)$ and $M(t)$, along with the result of (P1), seed downstream network inference. In the next stage, (P2) is used to estimate protein expression, and finally all obtained results are considered in algorithm 1 to produce a regulatory network map. Fig. 3 provides a graphical description of the bracketed pre-inference stages.

We impose the constraint that any given miRNA can only inhibit the expression of a particular target mRNA through one mode of regulation, either transcript cleavage or translational repression. This is reasonable, given that the particular pathway of inhibition is determined by the specificity of binding between a particular miRNA and a seed site on a target transcript, which is a fixed interaction for each miRNA-mRNA pairing [25], [26], [27]. This constraint takes the following mathematical form

$$\mathbb{I}_{\mathbb{R}^+}\{\lambda_{ij}^{RNA}\} + \mathbb{I}_{\mathbb{R}^+}\{\lambda_{ij}^{Prot}\} = 1.$$

3 Network Inference Algorithm

Sections 3.1, 3.2, 3.3, and 3.4 contain all the core algorithmic components in our proposed inference pipeline. A graphical overview of how these modular algorithms form a framework for gene network inference is shown in Fig. 2. This linear ordering of post-processing and inference steps, although designed for a normalized gene expression dataset involving a precise perturbation, is robust and flexible.

3.1 Modeling and Estimation of Gene Expression

Normalized gene expression values, such that $x_i(t) \leq 1$, are the given input for the algorithms described in this and subsequent sections. In reality, gene expression trajectories are inevitably noisy, which perturb the model parameters away from the true values. To reduce this noise effect, we first represent gene expressions as a linear combination of basis functions in the following form

$$x_i(t) = \sum_{d=1}^{D} \theta_{id} \varphi_d(t) = \boldsymbol{\varphi}(t)^T \boldsymbol{\theta}_i, \tag{6}$$

where D is the total number of bases and θ_{id} the coefficient of the dth basis function, $\varphi_d(t)$. The basis functions are chosen to take the form of a B-spline (Appendix B). Although all genes are associated with a common set of basis functions in (6), one can consider different sets of basis functions for different genes. Moreover, different basis functions such as a radial basis [28] or a mixed B-spline [29] may lead to more generalized characterization of gene expression. Nevertheless, we assume that gene trajectories smoothly change and a few B-spline basis functions are sufficient to estimate gene expressions.

The form of (6) allows us to fit a continuous function for a set of discrete gene expression measurements, using the following minimization

$$\text{(P1)} \qquad \min_{\boldsymbol{\theta}_i} \left\| \sum_{j=1}^{L} \left(x_i(t_j) - \boldsymbol{\varphi}(t_j)^T \boldsymbol{\theta}_i \right) \right\|_2 + \gamma_\theta \boldsymbol{\theta}_i^T \mathbf{K} \boldsymbol{\theta}_i,$$

where the roughness penalty $\boldsymbol{\theta}_i^T \mathbf{K} \boldsymbol{\theta}_i = \int_{t_1}^{t_L} \left(d^2 x_i(t)/dt^2 \right)^2 dt$ and $\mathbf{K}$ is a roughness matrix with the (j,k)th entry $\int_{t_1}^{t_L} \varphi_j''(t) \varphi_k''(t) dt$. Here, the first term is intended to diminish noise within measurements and the second term is intended to smooth our approximations. The parameter γ_θ is tuned by cross validation where training data is available, otherwise it can be drawn from a characterized network from the nearest available biological system.

Employing (P1), our estimation to $x_i(t)$, denoted as $\hat{x}_i(t)$, is a continuous function in time and its first derivative can be easily calculated as

$$\frac{d\hat{x}_i(t)}{dt} \simeq \frac{\hat{x}_i(t + \Delta t) - \hat{x}_i(t)}{\Delta t}. \tag{7}$$

Throughout the rest of the paper, it is assumed that our samples are taken from $\hat{x}_i(t)$ and therefore, any arbitrary number of samples, L, is achievable. We further replace $\hat{x}_i(t)$ with $x_i(t)$ for notational convenience.

3.2 Detection of Perturbed Genes

We can introduce a simple first approach for detecting when individual genes exit steady state post-perturbation. Gene expression models generated via (P1) are essentially smooth and noise-free when the total number of bases is restricted to an appropriately small number, D. High-frequency gene trajectories, whether a product of noise or periodicity in expression [30], [31], are converted into flat trajectories. This property allows us to detect when significant non-periodic deviations occur with respect to the initial steady state measurement(s). More precisely, time interval $[t_1, t_L)$ is divided into R sub-intervals as $[rt_{\{1,L\}}, (r+1)t_{\{1,L\}})$ for all $1 \leq r \leq R$, where $t_{\{1,L\}} := (t_1 - t_L)/(R+1)$. We choose R with respect to the nature of the original expression data, such that $R \geq D$.

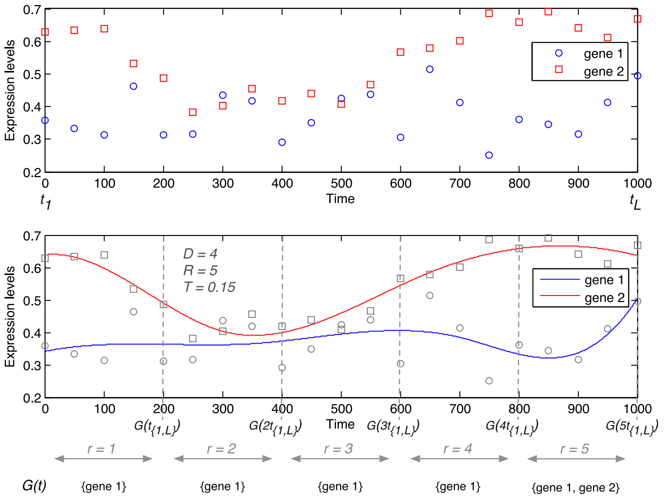


Fig. 3. The bracketed pre-inferenced stages of the pipeline in Fig. 2 are shown graphically. Discrete expression data from two genes and a small number of basis functions are utilized to produce continuous models of expression (P1), followed by segmentation and change detection. In this simple example, a change in gene 2 is detected in sub-interval $r = 1$, and a change in gene 1 is detected in sub-interval $r = 5$.

For each sub-interval, we look for the maximum and minimum values of trajectories. The sets $G(t)$ and $M(t)$ are then expanded as follows. At sub-interval r, gene i is included within either $G(t)$ or $M(t)$ for $t > rt_{\{1,L\}}$ provided that the deviation from the steady state measurement of gene i is greater than a desired threshold, T. In the simulations described in this paper, T was set in the range of $[0.15, 0.20]$ for normalized expression data. Both R and this threshold can be modified to better reflect the frequency of gene expression measurements for a given biological system (as shown in Fig. 3). If more complex change detection schemes are preferred, a number of alternative approaches can be adapted for this purpose [32], [33], [34].

3.3 Modeling and Estimation of Protein Expression

Formulation. Similar to (6), we express the protein level $y_i(t)$ as

$$y_i(t) = \sum_{d=1}^{D} \alpha_{id} \varphi_d(t) = \boldsymbol{\varphi}(t)^T \boldsymbol{\alpha}_i. \tag{8}$$

Our objective is first to find $\boldsymbol{\alpha}_i$ through the ODE (2) resulting in an estimation of the protein level $y_i(t)$. The calculated $y_i(t)$'s are in turn used to approximate unknown variables associated with the ODE (1). One of the challenges of solving non-linear ODEs is that the solution does not usually have a closed form. We propose to transform the non-linear ODE (2) into a linear regression problem. To motivate our method of constructing the ODE solution, we consider the first derivative of $y_i(t)$ as

$$y_i'(t) = \boldsymbol{\varphi}'(t)^T \boldsymbol{\alpha}_i,$$

and ODE (2) is consequently represented as

$$\boldsymbol{\varphi}'(t)^T \boldsymbol{\alpha}_i = \left(r_i - \sum_{j \in M(t)} \lambda_{ij}^{Prot} x_j(t) \right) x_i(t) - \lambda_i^{Prot} \boldsymbol{\varphi}(t)^T \boldsymbol{\alpha}_i.$$

We rewrite the above equation in the following form

$$r_i x_i(t) - \mathbf{x}_{M(t)}{}^T \boldsymbol{\lambda}_i^R(t) - \mathbf{b}_i(t)^T \boldsymbol{\alpha}_i = 0, \tag{9}$$

where $\mathbf{b}_i^T(t) := [\lambda_i^{Prot}\boldsymbol{\varphi}(t)^T + \boldsymbol{\varphi}'(t)^T]$ and $\boldsymbol{\lambda}_i^R(t)$ is the column vector with entries $\lambda_{ij}^{Prot}, \forall j \in M(t)$. The miRNA expressions corresponding to $\boldsymbol{\lambda}_i^R(t)$ are indicated by the vector $\mathbf{x}_{M(t)}$ such that both vectors, $\boldsymbol{\lambda}_i^R(t)$ and $\mathbf{x}_{M(t)}$, have the same index order. For notational convenience, we assume that all entries of $\mathbf{x}_{M(t)}$ are multiplied by $x_i(t)$.

Consider gene expressions at times t_l, $1 \leq l \leq L$. Setting all available gene expressions in equation (9), we arrive at

$$\mathbf{A}_i\left(-r_i, \mathbf{z}_i^T, \boldsymbol{\alpha}_i^T\right)^T = 0,$$

where

$$\mathbf{A}_i := \begin{pmatrix} x_i(t_1) & (\mathbf{x}_{M(t_1)}{}^T, \mathbf{0}(t_1)^T) & \mathbf{b}_i(t_1)^T \\ x_i(t_2) & (\mathbf{x}_{M(t_2)}{}^T, \mathbf{0}(t_2)^T) & \mathbf{b}_i(t_2)^T \\ \vdots & \vdots & \vdots \\ x_i(t_L) & \mathbf{x}_{M(t_L)}{}^T & \mathbf{b}_i(t_L)^T \end{pmatrix}, \quad \mathbf{z}_i := \boldsymbol{\lambda}_i^R(t_L),$$

and $\mathbf{0}(t_l)$ is the zero column vector with length $\mathbf{card}(M(t_L)) - \mathbf{card}(M(t_l))$. When the length is zero, we do not consider the vector $\mathbf{0}(t_l)$, e.g., $(\mathbf{x}_{M(t_L)}{}^T, \mathbf{0}(t_L)^T)$ is replaced by $\mathbf{x}_{M(t_L)}{}^T$ in the last row of $\mathbf{A}_i$. Matrix $\mathbf{A}_i$ has L rows and $\mathbf{card}(M(t_L)) + D + 1$ columns. Given that r_i is positive, we normalize $\left(-r_i, \mathbf{z}_i^T, \boldsymbol{\alpha}_i^T\right)^T$ with respect to r_i and represent the normalized vector as $\left(-1, \mathbf{z}_i^T, \boldsymbol{\alpha}_i^T\right)^T$, acknowledging abuse of notation. Given λ_i^{Prot} and $M(t)$, matrix $\mathbf{A}_i$ is completely determined.

Algorithm. We need to solve the linear system model

$$\mathbf{A}_i \begin{pmatrix} -1 \\ \mathbf{z}_i \\ \boldsymbol{\alpha}_i \end{pmatrix} = 0 \tag{10}$$

for $\mathbf{z}_i$ and $\boldsymbol{\alpha}_i$ when matrix $\mathbf{A}_i$ is determined. For identifiability of $\mathbf{z}_i$ and $\boldsymbol{\alpha}_i$, we require that $L \geq \mathbf{card}(M(t_L)) + D$, that is the number of equations is no smaller than the number of unknown parameters. However the sparsity in $\mathbf{z}_i$, given that only a small number of miRNAs typically act on a common gene [35], reduces the number of required equations.

To account for measurement noise and encourage $\mathbf{z}_i$ to be sparse, we will minimize the 2-norm error described in (10) with 1-norm regularization $\|\mathbf{z}_i\|_1$. Furthermore, we adopt the analogous roughness penalty $\boldsymbol{\alpha}_i^T\mathbf{K}\boldsymbol{\alpha}_i$ as used in (P1). Thus, we propose to obtain the ODE (2) solution with the following convex optimization

$$\text{(P2)} \qquad \min_{\{\mathbf{z}_i, \boldsymbol{\alpha}_i\}} \left\| \mathbf{A}_i \begin{pmatrix} -1 \\ \mathbf{z}_i \\ \boldsymbol{\alpha}_i \end{pmatrix} \right\|_2 + \gamma_z \|\mathbf{z}_i\|_1 + \gamma_\alpha \boldsymbol{\alpha}_i^T \mathbf{K} \boldsymbol{\alpha}_i,$$

$$\text{subject to} \quad \mathbf{z}_i \geq \mathbf{0}$$

$$(\mathbf{x}_{M(t_l)}{}^T, \mathbf{0}(t_l)^T)\mathbf{z}_i \leq x_i(t) \quad \forall 1 \leq l \leq L,$$

where γ_z and γ_α are chosen using cross validation. The second constraint ensures that the total rate of translation, $r_i - h_i(\mathcal{X}_{M(t)})$, is not negative. Due to the convex nature of this problem, it can be quickly solved for large gene datasets. This recovery of protein expression is dependent on prior knowledge of individual protein degradation rates, λ_i^{Prot}. In the absence of this experimental data, we can fix the value of λ_i^{Prot} to 1 for the entire system and still achieve accurate network reconstruction as shown in subsequent sections. While this assumption, combined with the normalization of $y_i(t)$ with respect to r_i, produces protein and gene expression levels in the same range of values, we may observe inaccuracies in inference resulting from the fact that true degradation and translation rates may occur on different scales.

3.4 Gene Regulatory Network Inference

Formulation. The model given by ODEs (1) and (2) describes the evolution of RNA and protein expressions provided that we know all the regulatory parameters, e.g., a_{ij}, b_{ij}, and τ_i. Coefficients a_{ij} and b_{ij} are difficult to experimentally determine and it is currently infeasible to carry out the relevant measurements simultaneously for a complex system with gene products under consideration. Our goal is to estimate these coefficients so that the ODE models can be temporally fitted to gene expression data. Specifically, we will use the previously described estimations of protein and RNA expression to approximate a_{ij} and b_{ij}, and to infer a regulatory network map.

To improve the reliability of the inferred network, we take into account time-dependent changes in gene levels and construct a set of equations accordingly. This is an important departure from standard steady state treatments. In this scenario, we first assume that the non-perturbed system is in an initial steady state, where RNA and protein levels are near constant (i.e., $dx_i(t)/dt = dy_i(t)/dt \simeq 0$). As previously mentioned, the perturbation of protein-encoding gene $x_i^p(t_1)$ first leads to fluctuations in the expression levels of genes in its immediate regulatory network. Genes that have exited a steady-state expression profile at any time up to t, $G(t)$ and $M(t)$, expand to contain greater numbers of genes that interact to form a putative regulatory network.

Considering changes in gene levels $x_i(t)$ at time t_l, $1 \leq l \leq L$, with the exception of $x_i^p(t_1)$, the term $\tau_i f_i(\mathcal{Y}_{G(t_l)})$ in equation (1) can be rewritten as follows

$$\tau_i f_i(\mathcal{Y}_{G(t_l)}) = \frac{\tau_i a_{i0} + \sum_{j=1}^{N(t_l)} \tau_i a_{ij} \prod_{k \in S_{ij}(t_l)} y_k(t_l)}{1 + \sum_{j=1}^{N(t_l)} b_{ij} \prod_{k \in S_{ij}(t_l)} y_k(t_l)} := \frac{\mathbf{p}_i^T(t_l)\mathbf{a}_i}{\mathbf{p}_i^T(t_l)\mathbf{b}_i}, \tag{11}$$

where $\mathbf{a}_i$ is a vector with $(j+1)$th entry $\tau_i a_{ij}$, $0 \leq j \leq N(t_L)$. The $(j+1)$th element of vector $\mathbf{p}_i(t_l)$ is described by $\prod_{k \in S_{ij}(t_l)} y_k(t_l)$ when $0 \leq j \leq N(t_l)$ and zero for $N(t_l) + 1 \leq j \leq N(t_L)$. Vector $\mathbf{b}_i$ is defined such that the first entry is 1 and $(j+1)$th, $1 \leq j \leq N(t_L)$, is b_{ij}.

Remark 1. Given that $y_i(t)$s are normalized with respect to r_i, a_{ij} and b_{ij} include the multiplier term $\prod_{k \in S_{ij}(t_l)} r_k$ so that the normalization can be vanished. Similarly, τ_i can be absorbed into the coefficients a_{ij}, where we assume $\tau_i < 1$ to maintain the algorithm constraint $0 \leq a_i \leq b_i$.

We also represent

$$\left(\lambda_i^{RNA} + \sum_{j \in M(t_l)} \lambda_{ij}^{RNA} x_j(t_l)\right) x_i(t_l) + \left.\frac{dx_i(t)}{dt}\right|_{t=t_l} := \mathbf{u}_i^T(t_l)\boldsymbol{\lambda}_i, \tag{12}$$

in which $\mathbf{u}_i(t_l)$ and $\boldsymbol{\lambda}_i$ are defined as follows. First and second entries of vector $\mathbf{u}_i(t_l)$ are $dx_i(t)/dt|_{t=t_l}$ and $x_i(t_l)$, respectively. The remaining entries are $x_j(t_l)x_i(t_l),\ j \in M(t_l)$. Making the same arrangement of array as $\mathbf{u}_i(t_l)$, vector $\boldsymbol{\lambda}_i$ is determined by first entry 1, second entry λ_i^{RNA}, and subsequent entries λ_{ij}^{RNA}, $j \in M(t_l)$.

Using (11)-(12), equation (1) can be reformulated as

$$\Omega_l(\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i) := \mathbf{p}_i^T(t_l)\mathbf{a}_i - \mathbf{u}_i^T(t_l)\boldsymbol{\lambda}_i \mathbf{b}_i^T \mathbf{p}_i(t_l) = 0. \tag{13}$$

Algorithm. We need to solve the non-convex problem

$$\begin{aligned} \text{(P3)} \qquad & \min_{\{\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i\}} \Gamma(\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i) \\ & \text{subject to} \quad 0 \le \mathbf{a}_i \le \mathbf{b}_i, 0 \le \boldsymbol{\lambda}_i \\ & \qquad\qquad\quad \mathbf{b}_i(1) = 1 \\ & \qquad\qquad\quad \boldsymbol{\lambda}_i(1) = 1, \boldsymbol{\lambda}_i(2) = \lambda_i^{RNA} \end{aligned}$$

with

$$\begin{aligned} \Gamma(\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i) := & \sum_{l=1}^{L} \Omega_l(\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i)^2 + \frac{\gamma_1}{2}\left(\|\boldsymbol{\lambda}_i\|_2^2 + \|\mathbf{b}_i\|_2^2\right) \\ & + \gamma_2\|\mathbf{b}_i\|_1 + \gamma_3\|\boldsymbol{\lambda}_i\|_1. \end{aligned}$$

The first term in the above equation follows from (13). The second term associated with $\gamma_1/2$ motivates grouping effect among variables $\mathbf{b}_i$ and $\boldsymbol{\lambda}_i$ [36], [37]. Due to the assumption that each gene has only a few regulators, 1-norm regularizations are considered to encourage sparse solutions. The validity of this assumption will vary based on the subnetwork and perturbation of interest. Immediacy to complex trans-acting regulators that act on many genes and networks is likely to confound local network recovery. Note that in the absence of miRNAs (all $\lambda_{ij}^{RNA} = 0$), terms $\|\boldsymbol{\lambda}_i\|_2$ and $\|\boldsymbol{\lambda}_i\|_1$ are no longer needed.

Non-convex optimizations are generally hard to solve in a reasonable time. Hence, we seek to identify a special treatment that reduces the computational complexity and provides desired solutions. Optimization (P3) is convex in $\{\mathbf{a}_i, \mathbf{b}_i\}$ for fixed $\boldsymbol{\lambda}_i$ and vice versa, and therefore the problem is bi-convex and can be solved using a variation of the alternating-direction method of multipliers (ADMM) which cycles over two groups of variables [38], cf. Appendix C. Here, since there are no equality constraints in (P3), ADMM is reduced to simple alternating minimization (see Section 9.2.1 in [39]). The proposed solver entails an iterative procedure, Algorithm 1, compromising two steps per iteration $k = 1, 2, \ldots$

The above iterative procedure implements a block coordinate descent method [40]. At each minimization, the variables that are not being updated are treated as fixed and are replaced with their most updated values. Then the iteration alternates between two sets of variables, $\{\mathbf{a}_i, \mathbf{b}_i\}$ and $\boldsymbol{\lambda}_i$.

One difficulty with the proposed solver is that it may result in stationary points which are not necessarily globally optimal. This occurs since optimization (P3) is not convex in $\{\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i\}$. Motivated by the proposition 1 in [41], the next theorem offers a global optimality certificate upon the convergence of the solver. In the case that the theorem is not applicable, we perform ADMM for multiple initials, $\{\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i\}$, and choose the local optima that leads to the smallest cost function in (P3).

Theorem 1. *Let $\{\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i\}$ be a stationary point of (P3). If*

$$\left\| \sum_{l=1}^{L} \Omega_l(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i)\mathbf{u}_i(t_l)\mathbf{p}_i^T(t_l) \right\| \le \frac{\gamma_1}{2}, \tag{14}$$

then $\{\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i\}$ is the globally optimal solution of (P3).

Proof. See Appendix D. □

Remark 2. For non-convex problems, ADMM offers no convergence guarantees. Nevertheless, there are evidences in the literature that show empirical convergence of ADMM, particularly when the non-convex exhibits specific structures. For example in our scenario, problem (P3) is bi-convex and admits unique closed form solutions for sub-problems **[S1]** and **[S2]**. This observation along with desired properties, [42, Theorems 4.5 and 4.9], are indeed a sufficient case for successful convergence. A formal proof of convergence is beyond the scope of this paper.

Algorithm 1 is intended for the case in which the RNA degradation rates, λ_i^{RNA}, are available. However, experimentally measuring λ_i^{RNA} is a difficult task. We offer a simple modification to the algorithm so that network inference can be still obtained without prior knowledge of RNA degradation rates.

Algorithm 1. Gene Regulatory Inference

```
input $\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i$
initialize $\mathbf{a}_i[0], \mathbf{b}_i[0]$, and $\boldsymbol{\lambda}_i[0]$ at random with respect to
    $\mathbf{b}_i(1) = 1, \boldsymbol{\lambda}_i(1) = 1, \text{and}\boldsymbol{\lambda}_i(2) = \lambda_i^{RNA}.$
for $k = 0, 1, \ldots$ do
  [S1] Update primal variables $\mathbf{a}_i$ and $\mathbf{b}_i$:
    $\{\mathbf{a}_i[k+1], \mathbf{b}_i[k+1]\} = \arg\min_{\{\mathbf{b}_i, \mathbf{a}_i\}} \Gamma(\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i[k])$
    subject to $\mathbf{0} \le \mathbf{a}_i \le \mathbf{b}_i$
               $\mathbf{b}_i(1) = 1.$
  [S2] Update primal variable $\boldsymbol{\lambda}_i$:
    $\boldsymbol{\lambda}_i[k+1] = \arg\min_{\boldsymbol{\lambda}_i} \Gamma(\mathbf{a}_i[k], \mathbf{b}_i[k], \boldsymbol{\lambda}_i)$
    subject to $\boldsymbol{\lambda}_i \ge \mathbf{0}$
               $\boldsymbol{\lambda}_i(1) = 1, \boldsymbol{\lambda}_i(2) = \lambda_i^{RNA}.$
end for
return $\mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\lambda}_i$
```

For simplicity of explanation, we can first remove miRNAs from our model. ODE (1) can then be rewritten as

$$\Omega_l(\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i) := \mathbf{p}_i^T(t_l)\left(\mathbf{a}_i - \mathbf{b}_i \frac{dx_i(t_l)}{dt} - \mathbf{c}_i x_i(t_l)\right) = 0, \tag{15}$$

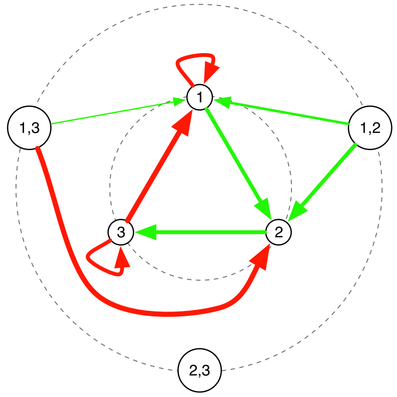


Fig. 4. Map of gene regulatory network described by equations (17) and (18) . First-order (single) and and second-order (combined) regulators are depicted in concentric circles. Green arrows specify gene activation and red arrows specify gene repression. The relative magnitudes of activation and repression are roughly represented by arrow thickness.

and $\mathbf{c}_i := \lambda_i^{RNA}\mathbf{b}_i$. Employing the above reformulation, unknown variables $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{c}_i$ are estimated through the following convex optimization

$$
\begin{aligned}
\text{(P4)}\quad \arg\min_{\{\mathbf{a}_i,\mathbf{b}_i,\mathbf{c}_i\}} \; & \sum_{l=1}^{L}\Omega_l(\mathbf{a}_i,\mathbf{b}_i,\mathbf{c}_i) + \gamma_2(\|\mathbf{b}_i\|_1 + \|\mathbf{c}_i\|_1) \\
\text{subject to} \quad & \mathbf{0} \le \mathbf{a}_i \qquad (16) \\
& \mathbf{a}_i \le \mathbf{b}_i \\
& \lambda_{min}\mathbf{b}_i \le \mathbf{c}_i \le \lambda_{max}\mathbf{b}_i,
\end{aligned}
$$

where λ_{min} and λ_{max} specify an lower and upper bound for λ_i^{RNA}, respectively. Variable $\mathbf{c}_i$ is introduced to remove λ_i^{RNA} from our optimization. However, the new variable expands the feasible set of solutions, which might create an answer different from the true value. To reduce this effect, we add constraint (16) to (P4) to tighten the feasible set of solutions. Given that $\lambda_i^{RNA}/\tau_i \ge 1$, we can take on the additional constraint $\mathbf{a}_i \le \mathbf{c}_i$. In the subsequent simulations, λ_{min} is in the near-zero range $[0.001, 0.01]$, and λ_{max} is selected in the range $[0.1, 1]$. It is straightforward to generalize the introduced approach within the framework of (P3). Derivations are removed to avoid repetition in the paper.

4 Simulations

4.1 Small Gene Network with Prior Knowledge of Degradation Rates

To demonstrate the proposed time-series approach, we consider the three-gene network described by the following systems of ODEs for gene expression

$$
\begin{aligned}
\frac{dx_1(t)}{dt} =& \frac{0.1 + 0.05y_1(t)y_2(t) + 0.025y_1(t)y_3(t)}{1 + 0.1y_1(t) + 10y_3(t) + 0.05y_1(t)y_2(t) + 0.025y_1(t)y_3(t)}, \\
& - 0.1x_1(t), \\
\frac{dx_2(t)}{dt} =& \frac{0.1 + 0.1y_1(t) + 0.1y_1(t)y_2(t)}{1 + 0.1y_1(t) + 0.1y_1(t)y_2(t) + 10y_1(t)y_3(t)} - 0.1x_2(t), \\
\frac{dx_3(t)}{dt} =& \frac{0.1 + 0.1y_2(t)}{1 + 0.1y_2(t) + .1y_3(t)} - 0.1x_3(t),
\end{aligned}
\qquad (17)
$$

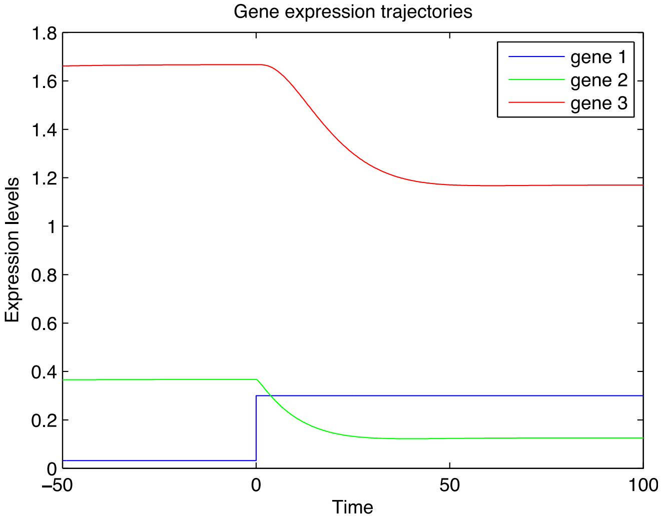


Fig. 5. Gene expression trajectories (unnormalized) before and during the imposed perturbation. The system is in steady state before time 0. Gene 1 is artificially perturbed at time zero, leading to changes in gene expression levels. A new steady state is eventually achieved at approximately time 50. We sample expression levels between time 0 (the starting point of perturbation) and 50 (the new steady state) and use them as data in our algorithm.

and the following system of ODEs for protein expression

$$
\begin{aligned}
\frac{dy_1(t)}{dt} =& x_1(t) - 0.5y_1(t), \\
\frac{dy_2(t)}{dt} =& 2x_2(t) - 0.5y_2(t), \qquad (18) \\
\frac{dy_3(t)}{dt} =& x_3(t) - 0.5y_3(t).
\end{aligned}
$$

The above toy model, visualized in Fig. 4, is provided to better explain our algorithms. The explicit system of ODEs, describing the kinetics of the system [43], allows us to generate samples to fit our model and to also compare recovered solutions with the ground truth. This model also incorporates complex modes of regulation, including self-regulation and combined regulators.

To generate data, arbitrary initial conditions are assigned to ODEs (17) and (18) and the system is allowed to resolve to a steady state. To perturb this steady state, the expression level of gene 1, $x_1(t)$, is artificially fixed to 0.3, leading to fluctuations in the expression levels of other genes. Fig. 5 illustrates expression trajectories before and during the perturbation.

We collect 12 samples from each gene expression level. The samples are chosen uniformly from time interval $[0, 50]$. Points 0 and 50 specify the times at which the perturbation starts and the system reaches a new steady state, respectively. Using these sampled data, we solve optimization (P2) to effectively recover protein expressions as shown in Fig. 6.

We finally examine Algorithm 1, (P3), for the goal of network recovery. In this scenario, our target is to estimate vectors $\mathbf{a}_i$ and $\mathbf{b}_i$. We assume that the degradation rates are known in advance and therefore, since the system does not contain any miRNA in this particular example, $\boldsymbol{\lambda}_i$ is completely at hand. Let us consider gene 3 where the true value of $\mathbf{a}_3 = (0.1, 0, 0.1, 0, 0, 0, 0)$ and $\mathbf{b}_3 = (1, 0, 0.1, 0.1, 0, 0, 0)$. Vectors $\mathbf{a}_3$ and $\mathbf{b}_3$ are indexed with regard to

$$\mathbf{p}_3(t_l) = (1, y_1(t_l), y_2(t_l), y_3(t_l), y_1(t_l)y_2(t_l), y_1(t_l)y_3(t_l), y_2(t_l)y_3(t_l)).$$

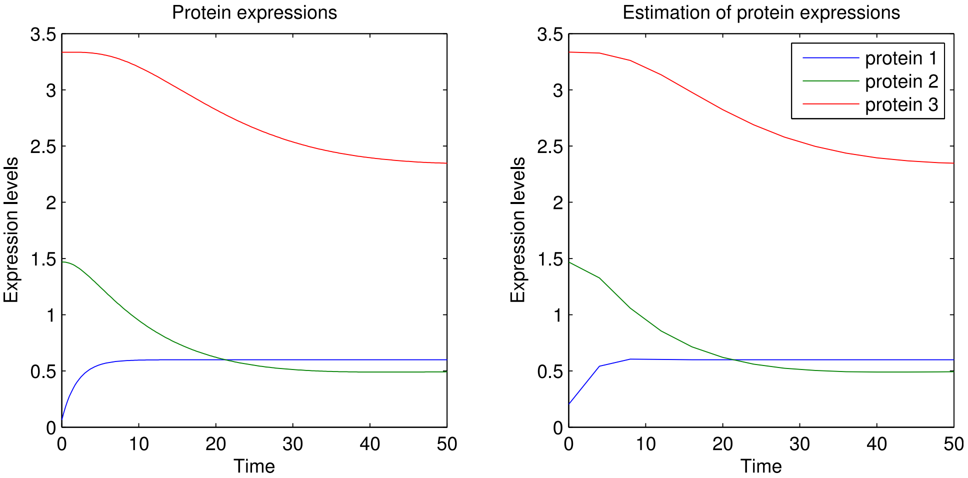


Fig. 6. Exact protein expression curves derived from model ODEs (17) and (18) (left), and their recovered estimations using 12 unnormalized time-point samples via (P2) (right). For convenience of graphical comparison, the values of r_i were drawn from the system equations. Protein expression is otherwise normalized with respect to r_i, but this would result in a transformed scale for this qualitative comparison.

Applying our method, we obtain $\mathbf{a}_3 \simeq (0.1, 0, 0.083, 0, 0, 0, 0)$ and $\mathbf{b}_3 \simeq (1, 0, 0.083, 0.08, 0, 0, 0)$. Table 1 demonstrates that as the sampling frequency increases, we attain more accurate approximations. Furthermore, it can be seen that the estimations achieve similar accuracy after a small number of samples.

Employing the aforementioned single perturbation, we are only able to recover the strongest edge of gene 2, $\mathbf{b}_2(6) = 10$. The difficulty here is due to the sharp change in y_1 (Fig. 6), which provides us with a minimal amount of dynamic information. y_1 near-instantaneously switches between two steady-state levels of expression, resulting in less accurate recovery of the underlying dynamics. However, expression patterns in perturbed biological settings tend to be more dynamic and are unlikely to contain this type of expression pattern. In this example, the removal of sharp instantaneous expression changes leads to complete recovery of the gene regulatory network.

Remark 3. The recovery of regulatory networks using this proposed approach is tightly associated with the presence of dynamic changes in gene expression. These changes can provide us with a certain amount of information which predominantly specifies the accuracy of estimation. The achievable accuracy depends on many factors such as nonlinearity in changes or similarity in the range of changes.

TABLE 1
Inference of Binding Coefficients Describing Energies of Regulator Complex-Promoter Interactions Based on Number of Samples

# of Samples	Variables	Estimated vector entries						
8	$\mathbf{a}_3$	0.097	0	0.082	0	0	0	0
	$\mathbf{b}_3$	1	0	0.082	0.06	0	0	0
16	$\mathbf{a}_3$	0.1	0	0.093	0	0	0	0
	$\mathbf{b}_3$	1	0	0.093	0.089	0	0	0
24	$\mathbf{a}_3$	0.1	0	0.1	0	0	0	0
	$\mathbf{b}_3$	1	0	0.1	0.092	0	0	0

System identification for gene 3 based on sample frequency. Coefficients $\mathbf{a}_3$ and $\mathbf{b}_3$ are the numerator and denominator of p_3^{bound} (binding probability of RNA polymerase to a given promoter).

4.2 Medium (10-Gene) Simulated Network with Noise

We extend our approach to simulated networks of 10 genes, generated as part of the DREAM4 *in silico* network inference challenge [45]. The available data for each network includes a simulated time series of gene expression in response to five individual chemical perturbations, along with single steady-state expression levels for wild-type, knockdown, knockout, and multifactorial perturbations. These datasets also simulate internal network noise and incorporate measurement noise. We use these data to assess the robustness of our approach in a non-ideal setup.

Our approach is geared towards precise genetic and chemical perturbations, while these datasets simulate chemicals that are non-specific (NS) in their interactions. To place us at further disadvantage, we attempt network recovery using only the time series perturbations, forgoing all other datasets available to solvers. Lastly, our approach works best under conditions where RNA and protein degradation rates are known. Given that this information is unavailable, this exercise also serves as a test of our simplifying assumptions for such situations. Unlike simulations in the previous section, the rules of this challenge stipulate no self-regulation and no combined regulators.

DREAM4 Challenge 2 datasets for Networks 1 and 2 are used to infer gene regulatory networks and to inspect predictions of network topology using the official scoring pipeline. First, we use (P1) to produce smooth and continuous gene expression trajectories from the discrete and noisy time series dataset (Fig. 7). We are given such datasets each associated with an individual chemical perturbation. For each dataset, perturbed genes are identified and incorporated as described in Section 3.2. Network inference is carried out using Algorithm 1 and the inferred networks corresponding to five chemical perturbations are joined as one network. In the absence of RNA degradation rates, λ_{min} is set to either 0.001 or 0.01, and λ_{max} is set to 0.1 or 1. If a directed network edge is identified, the probability of the edge is set to 1 for weighted edges, and 0 otherwise. This is done to allow scoring of our network with the provided scripts, given our non-probabilistic formulation. Algorithm 1 minimization values are filtered against abnormal values that could represent underfitting and overfitting of data.

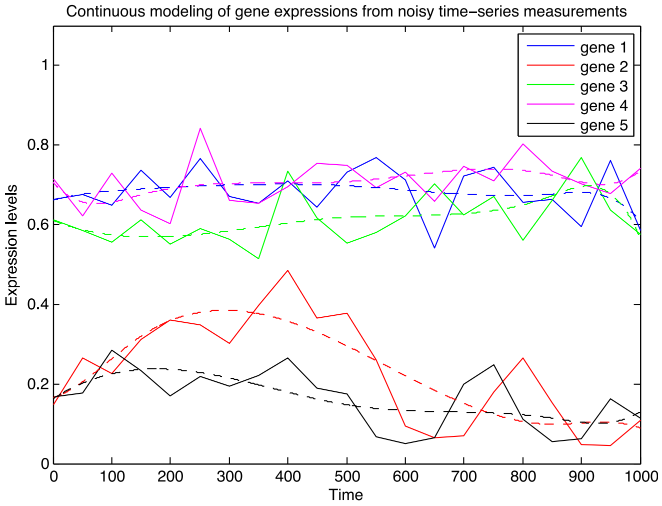


Fig. 7. Time series gene expression measurements from simulated DREAM4 datasets are shown with connected solid lines. Dashed lines of corresponding color show that application of (P1) effectively produces noise-free (smooth) and continuous gene expression curves.

For Network 1, we report the area under the receiver operating characteristic curve (AUROC) $= 0.81$ and the area under the precision-recall curve (AUPR) $= 0.75$, and for Network 2, AUROC $= 0.76$ and AUPR $= 0.68$. These results compare very favorably to other time series-based methods applied to the same datasets [46]. In fact, for Networks 1 the best AUROC and AUPR values that can offered by all approaches in [46] are 0.78 and 0.64, and for Network 2, 0.75 and 0.54. Therefore, our AUROC and AUPR values shows improvements over the top reported results.

4.3 Network Inference from Yeast Cell Cycle Time Series

In order to probe real biological data with inherent noise, we apply parts of our pipeline to a classical yeast cell cycle microarray dataset [47]. This data is provided as a 25 point time-series with a 5 minute sampling interval. Given the yeast cell is in an incredibly dynamic stage post synchronization with α-factor pheromone, this again represents a vast departure from ideal near steady-state conditions with a precise and local perturbation. We chose to focus our analysis on a set of primary regulatory genes and complexes involved in core cell cycle control and that showed greater than 15 percent changes in expression over the time course [44]. This led to retainment of seven genes. We use (P2) to fit smooth continuous functions to the noisy gene expression measurements exhibited in Fig. 8. We next examine our proposed scheme, (P4), to infer a gene regulatory network among these genes.

The inferred network is shown in Fig. 8, with arrows indicating directed edges for gene-gene excitatory and inhibitory interactions. Of the 12 regulatory interactions inferred, six are correct in both directionality and influence (i.e., inhibition versus activation) and two are correct only in directionality. Further, three can be considered conditionally correct, whereby the predicted influence is mediated by a single intermediate node that was absent from the model. A single edge was labeled as a false positive, even though an argument can be made for mediation of that influence by two intermediate nodes. Only two known regulatory edges were not recovered with our algorithm (SWI5 is a positive regulator of SIC1, and FKH2 is a positive regulator of SWI5 [44]). Strikingly, the algorithm correctly predicts a role for combined regulators and recovers the only example of self-regulation in the reference pathway. This is promising, given the absence of data relating to protein degradation, contextless inference, and the non step-wise nature of changes in expression that would be preferred in our proposed experimental scheme.

5 Conclusions

The gene inference pipeline described in this work helps establish a robust framework for network discovery from perturbed expression data. The system of equations used to model eukaryotic gene regulation include the novel extension of a thermodynamic and statistical mechanic approach to polymerase binding. This pipeline is best suited for the processing of expression measurements from high-resolution time series experiments involving precise genetic or chemical perturbation of a steady state system. Although our approach

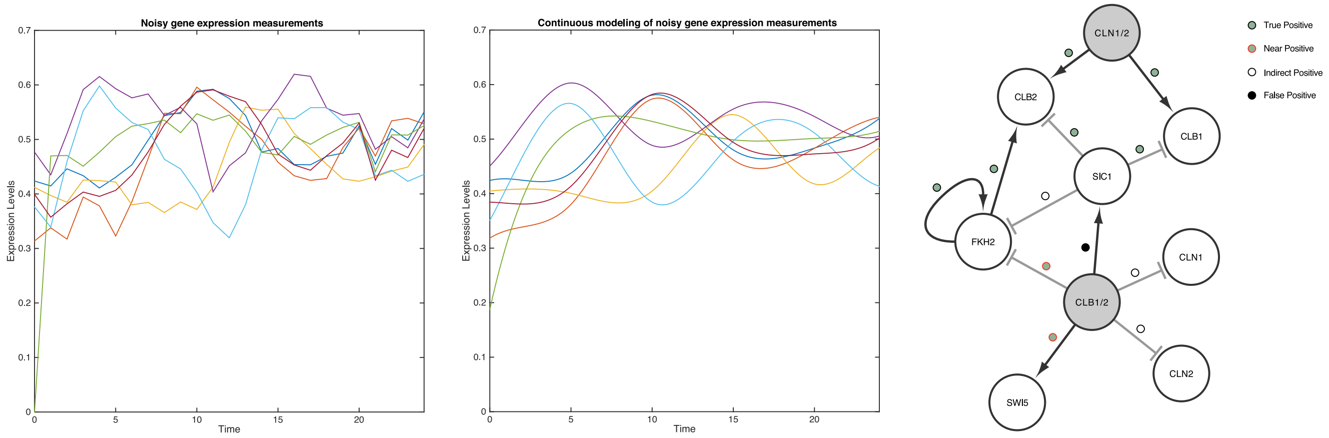


Fig. 8. Time-series gene expression measurements of yeast cell cycle-associated genes filtered at a stringent change detection threshold ($T = 0.15$) (left), and their recovered estimations using (P2) (center). The inferred network via (P4) is shown on the right, compared to the network as it's presently understood ([44]). "True positives" represent edges recapitulated by the inference algorithm in both direction and influence, "near positives" represent edges correct in direction but with reversed influence, "indirect positives" represent edges of correct direction and influence with a missing intermediate node, and "false positive" indicates an edge not found in the reference network and that cannot be explained through a single intermediate node.

aims for small network recovery, we expect that reliable network inference at larger scales can be achieved with data from multiple individual perturbations acting within a given network. Genetic perturbation is best in the form of induced over-expression or RNAi-mediated gene knockdown. Chemical perturbation is best in the form of a chemical that has a specific protein interaction and limited off-target effects. However, we establish that this approach can yield insights under non-ideal conditions.

The modular nature of our pipeline allows for the modification of different stages to best fit a given biological system and of expression information. Alternative approaches can be implemented for the stages that precede the core inference algorithm, including change detection. The performance of this approach can further be improved with *a priori* knowledge of protein expression levels, protein and RNA degradation rates, along with the labeling of noncoding RNAs. Technologies are continually being improved for the purpose of capturing these data in a genome-wide manner [48], [49], [50], [51], to complement gene expression measurements. Our gene inference approach can readily utilize protein expression data, protein and RNA degradation data, and miRNA labeling data.

While we expect such inference approaches to work better for homogenous and synchronized single-cell or single-tissue systems, we also expect to capture the most prominent and meaningful aspects of the aggregate dynamics of heterogenous mixed-cell populations, multi-tissue systems, and whole organisms. Future directions include the more comprehensive validation and refinement of these algorithms for synthetic networks and higher-order eukaryotic systems, adaptations of more sophisticated change detection schemes, and surveys of a broader range of system-specific sampling frequencies.

This inference method has broad application in biological network discovery. For example, it can be used to identify the topology of gene regulatory networks immediate to drug response, and can be used to identify new interactions for genes implicated in disease. The inference data can then be used to seed and prioritize candidates for downstream biological and *in vivo* validation.

APPENDIX

A. Treatment of Protein Regulators

Consider a gene for which the probability of RNAP being bound to a specific promoter site, S, is under the potential influence of a single non-steady state regulator, Regulator 1, and the collection of all available regulators still in steady state. The steady state regulators are encapsulated as a single super-protein complex, SS, that is fixed as bound to the promoter region. Suppose that we have P RNAP, R_1 Regulator 1, and R_{SS} super-protein complex.

We apply the following notation: ε_P^{NS} is used to denote the energy of the case in which RNAP is bound to a nonspecific DNA binding site, $\varepsilon_{P,i0}^{S}$ the energy when RNAP is only bound to the S binding site, $\varepsilon_{P,i1}^{S}$ the energy when RNAP is specifically bound to the promoter-regulator complex, ε_{SS}^{NS} the energy when the SS is bound to the NS binding site, ε_{SS}^{S} the energy when the SS is bound to the S binding site, ε_{i1}^{NS} the energy when Regulator 1 is bound to the NS binding site, ε_{i1}^{S} the energy when Regulator 1 is bound to the S binding site, and

$$\Delta\varepsilon_{P,i0} := \varepsilon_{P,i0}^{S} - \varepsilon_P^{NS}, \Delta\varepsilon_{P,i1} := \varepsilon_{P,i1}^{S} - \varepsilon_P^{NS}, \Delta\varepsilon_{i1} := \varepsilon_{i1}^{S} - \varepsilon_{i1}^{NS}.$$

Also define

$$Z(P, R_1, R_{SS}-1) := \frac{m! e^{-P\beta\varepsilon_P^{NS}} e^{-R_1\beta\varepsilon_{i1}^{NS}} e^{-(R_{SS}-1)\beta\varepsilon_{SS}^{NS}} e^{-\beta\varepsilon_{SS}^{S}}}{P! R_1! (R_{SS}-1)! (m-P-R_1-R_{SS}+1)!},$$

where $Z(P, R_1, R_{SS}-1)$ gives the total number of arrangements for RNAP and R1 at NS binding sites, weighted by a Boltzmann factor providing a relative energy for each state.

The available configurations of the system with corresponding unnormalized probabilities are enumerated as follows: (i) Regulator 1 and RNAP unbound: $Z(P, R_1, R_{SS}-1)$, (ii) only Regulator 1 bound: $Z(P, R_1-1, R_{SS}-1)e^{-\beta\varepsilon_{i1}^{S}}$, (iii) only RNAP bound: $Z(P-1, R_1, R_{SS}-1)e^{-\beta\varepsilon_{P,i0}^{S}}$, and (iv) both Regulator 1 and RNAP bound: $Z(P-1, R_1-1, R_{SS}-1)e^{-\beta\varepsilon_{P,i1}^{S}}$. To derive the probability of RNAP binding, we sum the probabilities of configurations in which RNAP is bound to the specific site and divide over the sum of probabilities of all potential configurations, Z_{total}. Here, in parallel to [22], it is shown how the effect of steady state proteins can effectively be removed from the protein regulator formulation, under the aforementioned arrangement. To represent the probability of RNAP binding to the cis regulatory region of gene i, we define p_i^{bound} as follows:

$$\begin{aligned} p_i^{\text{bound}} &= \Bigg(Z(P-1, R_1, R_{SS}-1)e^{-\beta\varepsilon_{P,i0}^{S}} \\ &\quad + Z(P-1, R_1-1, R_{SS}-1)e^{-\beta\varepsilon_{P,i1}^{S}} \Bigg) \Bigg/ Z_{\text{total}} \\ &= \Bigg(Z(P-1, R_1, R_{SS}-1)e^{-\beta\varepsilon_{P,i0}^{S}} \\ &\quad + Z(P-1, R_1-1, R_{SS}-1)e^{-\beta\varepsilon_{P,i1}^{S}} \Bigg) \Bigg/ \\ &\quad \Bigg(Z(P, R_1, R_{SS}-1) + Z(P, R_1-1, R_{SS}-1)e^{-\beta\varepsilon_{i1}^{S}} \\ &\quad + Z(P-1, R_1, R_{SS}-1)e^{-\beta\varepsilon_{P,i0}^{S}} \\ &\quad + Z(P-1, R_1-1, R_{SS}-1)e^{-\beta\varepsilon_{P,i1}^{S}} \Bigg) \\ &= \left(\frac{m}{R_1} e^{\beta\varepsilon_P^{NS}} e^{-\beta\varepsilon_{P,i0}^{S}} + e^{\beta\varepsilon_P^{NS}} e^{\beta\varepsilon_{i1}^{NS}} e^{-\beta\varepsilon_{P,i1}^{S}} e^{-\beta\varepsilon_{i1}^{S}} \right) \\ &\quad \Bigg/ \Bigg(\frac{m}{R_1} e^{\beta\varepsilon_P^{NS}} e^{-\beta\varepsilon_{P,i0}^{S}} + e^{\beta\varepsilon_P^{NS}} e^{\beta\varepsilon_{i1}^{NS}} e^{-\beta\varepsilon_{P,i1}^{S}} e^{-\beta\varepsilon_{i1}^{S}} \\ &\quad + \frac{m^2}{PR_1} + \frac{m}{P} e^{\beta\varepsilon_{i1}^{NS}} e^{-\beta\varepsilon_{i1}^{S}} \Bigg) \\ &= \frac{\frac{1}{y_1} e^{-\beta\Delta\varepsilon_{P,i0}} + e^{-\beta\Delta\varepsilon_{P,i1}} e^{-\beta\Delta\varepsilon_{i1}}}{\frac{1}{y_1} e^{-\beta\Delta\varepsilon_{P,i0}} + e^{-\beta\Delta\varepsilon_{P,i1}} e^{-\beta\Delta\varepsilon_{i1}} + \frac{1}{Py_1} + \frac{1}{P} e^{-\beta\Delta\varepsilon_{i1}}} \\ &= \frac{Pe^{-\beta\Delta\varepsilon_{P,i0}} + y_1 Pe^{-\beta\Delta\varepsilon_{P,i1}} e^{-\beta\Delta\varepsilon_{i1}}}{Pe^{-\beta\Delta\varepsilon_{P,i0}} + y_1 Pe^{-\beta\Delta\varepsilon_{P,i1}} e^{-\beta\Delta\varepsilon_{i1}} + 1 + y_1 e^{-\beta\Delta\varepsilon_{i1}}} \\ &= \frac{Pe^{-\beta\Delta\varepsilon_{P,i0}} + y_1 Pe^{-\beta\Delta\varepsilon_{P,i1}} e^{-\beta\Delta\varepsilon_{i1}}}{(1 + Pe^{-\beta\Delta\varepsilon_{P,i0}}) + y_1 e^{-\beta\Delta\varepsilon_{i1}} (1 + Pe^{-\beta\Delta\varepsilon_{P,i1}})}, \end{aligned}$$

where we have applied the approximation $m!/P!R_1!\,(R_{SS}-1)!(m-P-R_1-R_{SS}+\ 1)! \approx m^P m^{R_1} m^{R_{SS}}/P!R_1!(R_{SS}-1)!$. We introduce y_1, the protein product of Regulator 1 defined as R_1/m, for the purposes of normalization and in keeping with the protein designations used throughout this paper. We additionally note that P in the final steps of the derivation above is also normalized to m, but we retain the same notation for simplicity.

The final derivation can be generalized to, for an indefinite number of first and second order regulators

$$f_i(\mathcal{Y}_{G(t)}) = \frac{\sum_{j=0}^{N(t)} Pe^{-\beta\Delta\varepsilon_{P,ij}} e^{-\beta\Delta\varepsilon_{ij}} \prod_{k\in S_{ij}(t)} y_k(t)}{\sum_{j=0}^{N(t)} (1+Pe^{-\beta\Delta\varepsilon_{P,ij}}) e^{-\beta\Delta\varepsilon_{ij}} \prod_{k\in S_{ij}(t)} y_k(t)}, \quad (19)$$

where $\Delta\varepsilon_{ij}$ is the binding energy of the jth complex to the promoter, $\Delta\varepsilon_{P,ij}$ is the energy of RNAP being bound to the promoter-regulator complex j, and P is the concentration of RNAP. Setting $a_{ij} = Pe^{-\beta\Delta\varepsilon_{P,ij}} e^{-\beta\Delta\varepsilon_{ij}}$ and $b_{ij} = (1+Pe^{-\beta\Delta\varepsilon_{P,ij}})e^{-\beta\Delta\varepsilon_{ij}}$, we arrive at the form given in (3).

B. B-Splines

B-splines have been well investigated in approximation theory and numerical analysis, leading to a variety of important properties such as computational efficiency and numerical stability. Particularly, the B-spline basis functions have the best approximation capacity based on the Stone-Weierstrass Approximation Theorem. Polynomial functions are also used to estimate continuous functions. However, the B-spline bases are shown to be optimally stable [52].

A set of B-spline basis functions in variable t is determined by the degree of a piecewise polynomial, P, and a knot sequence [53]. The knot sequence is a set of points that divides a real interval into a number of sub-intervals. More precisely, D bases of degree P are parameterized by $D+P+1$ knots, $\{t_0, t_1, \ldots, t_{D+P}\}$ where $t_0 \le t_1 \le \ldots \le t_{D+P}$. Employing this set of knots and the De Boor recursion in [54], the dth B-spline basis of degree P, written as $\varphi_d^{(P)}(t)$, is derived recursively as follows:

$$\varphi_d^{(0)}(t) = \begin{cases} 1 & \text{if } t_{d-1} \le t \le t_d \\ 0 & \text{if otherwise} \end{cases}, \quad (20)$$

$$\varphi_d^{(p)}(t) = \frac{t-t_{d-1}}{t_{p+d-1}-t_{d-1}}\varphi_d^{(p-1)}(t) + \frac{t_{p+d}-t}{t_{p+d}-t_d}\varphi_{d+1}^{(p-1)}(t), \quad (21)$$

for $1 \le d \le D+P-p$ where $p=0$ in (20) and $1 \le p \le P$ in (21). The above recursion is visualized in Fig. 9 (reconstructed from [53]).

The degree $P=3$ or 4 is sufficient in most applications. The number of basis functions should be large enough to arrive at accurate estimation but not too large to cause overfitting. In our case, gene and protein levels do not contain high frequency changes and therefore, a small number of basis functions are sufficient to represent gene and protein expressions.

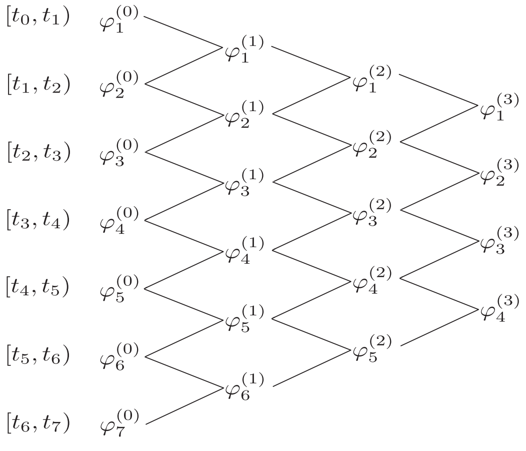


Fig. 9. The De Boor recursion for $P=3$ and $D=4$.

C. Bi-Convex Problems

Bi-convex optimization is a generalization of convex optimization where the objective function and the constraint set can be bi-convex [42].

Definition 1. *Let $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ be two non-empty convex sets. The set $B \subseteq \mathcal{X} \times \mathcal{Y}$ is called bi-convex if $B_x := \{y \in \mathcal{Y} : (x,y) \in B\}$ is convex for each x, and $B_y := \{x \in \mathcal{X} : (x,y) \in B\}$ is convex for each y.*

Definition 2. *A function $f(x,y) : B \to \mathbb{R}$ is called bi-convex if $f(x,y)$ is convex on B_x for every fixed x and also convex on B_y for every fixed y.*

A common method to solve a bi-convex problem is ADMM [39]. The ADMM is an iterative augmented Lagrangian method that uses partial updates for dual variables and replaces joint minimization by simpler subproblems. However, the mentioned procedure does not guarantee global optimality of the solution.

D. Proof of Theorem 1

The stationary points $\{\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i\}$ of (P3) are derived by setting sub-gradients to zero as follows

$$\nabla_{\mathbf{a}_i}\Gamma(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i) = 2\sum_{l=1}^{L} \Omega_l(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i)\mathbf{p}_i(t_l) = \mathbf{0}, \quad (22)$$

$$\begin{aligned}\nabla_{\mathbf{b}_i}\Gamma(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i) = -2\sum_{l=1}^{L} \Omega_l(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i)\mathbf{u}_i^T(t_l)\bar{\boldsymbol{\lambda}}_i\mathbf{p}_i(t_l) \\ + \gamma_1\bar{\mathbf{b}}_i + \gamma_2\,\mathrm{sign}(\bar{\mathbf{b}}_i) = \mathbf{0},\end{aligned} \quad (23)$$

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}_i}\Gamma(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i) = -2\sum_{l=1}^{L} \Omega_l(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i)\mathbf{p}_i^T(t_l)\bar{\mathbf{b}}_i\mathbf{u}_i(t_l) \\ + \gamma_1\bar{\boldsymbol{\lambda}}_i + \gamma_3\,\mathrm{sign}(\bar{\boldsymbol{\lambda}}_i) = \mathbf{0}\end{aligned} \quad (24)$$

with respect to constraints $0 \le \bar{\mathbf{a}}_i \le \bar{\mathbf{b}}_i$ and $\bar{\boldsymbol{\lambda}}_i \ge 0$. These constraints admit that $\mathrm{sign}(\cdot)$ can be replaced by vector $\mathbf{1}$ in the above equations. It is obvious from (23)-(24) that $\bar{\mathbf{b}}_i^T\nabla_{\mathbf{b}_i}\Omega(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i) = \bar{\boldsymbol{\lambda}}_i^T\nabla_{\boldsymbol{\lambda}_i}\Omega(\bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\boldsymbol{\lambda}}_i) = 0$, which results in

$$\begin{aligned}2\sum_{l=1}^{L}\Omega_l(\bar{\mathbf{a}}_i,\bar{\mathbf{b}}_i,\bar{\boldsymbol{\lambda}}_i)\mathbf{p}_i^T(t_l)\bar{\mathbf{b}}_i\mathbf{u}_i^T(t_l)\bar{\boldsymbol{\lambda}}_i &= \gamma_1\bar{\mathbf{b}}_i^T\bar{\mathbf{b}}_i+\gamma_2\bar{\mathbf{b}}_i^T\mathbf{1}\\ &= \gamma_1\bar{\boldsymbol{\lambda}}_i^T\bar{\boldsymbol{\lambda}}_i+\gamma_3\bar{\boldsymbol{\lambda}}_i^T\mathbf{1}.\end{aligned} \tag{25}$$

Consider the convex optimization

$$\begin{aligned}\text{(P5)}\quad \min_{\{\mathbf{a}_i,\mathbf{G}_i,\mathbf{W}_1,\mathbf{W}_2\}} &\sum_{l=1}^{L}\big(\mathbf{p}_i^T(t_l)\mathbf{a}_i-\mathbf{u}_i^T(t_l)\mathbf{G}_i\mathbf{p}_i(t_l)\big)^2\\ &+\gamma_1\kappa(\mathbf{W}_1,\mathbf{W}_2)\\ \text{subject to}\quad &\mathbf{W}:=\begin{pmatrix}\mathbf{W}_1 & \mathbf{G}_i\\ \mathbf{G}_i^T & \mathbf{W}_2\end{pmatrix}\succeq 0,\end{aligned} \tag{26}$$

where $\kappa(\mathbf{W}_1,\mathbf{W}_2):=\frac{1}{2}(\mathrm{Tr}(\mathbf{W}_1)+\mathrm{Tr}(\mathbf{W}_2))$.

Minimizing (P5) with respect to $\{\mathbf{W}_1,\mathbf{W}_2\}$ leads to

$$\|\mathbf{G}_i\|_*=\min_{\{\mathbf{W}_1,\mathbf{W}_2\}}\kappa(\mathbf{W}_1,\mathbf{W}_2)\quad\text{subject to}\quad\mathbf{W}\succeq 0,$$

which is the alternative characterization of the nuclear norm [55]. Taking advantage of the nuclear norm, we can restrict matrix $\mathbf{G}_i$ to be rank one as $\boldsymbol{\lambda}_i\mathbf{b}_i^T$. Also, $\kappa(\cdot,\cdot)$ is able to satisfy the required sparsity for $\{\boldsymbol{\lambda}_i,\mathbf{b}_i^T\}$. To investigate these claims, recall constraints in (P3) and set $\mathbf{G}_i:=\boldsymbol{\lambda}_i\mathbf{b}_i^T$, $\mathbf{W}_1:=\boldsymbol{\lambda}_i\boldsymbol{\lambda}_i^T+\frac{\gamma_3}{\gamma_1}\,\mathrm{diag}(\boldsymbol{\lambda}_i)$, and $\mathbf{W}_2:=\mathbf{b}_i\mathbf{b}_i^T+\frac{\gamma_2}{\gamma_1}\,\mathrm{diag}(\mathbf{b}_i)$ where $\mathrm{diag}(\boldsymbol{\lambda}_i)$ is the diagonal matrix with (j,j)th entry equal to $\boldsymbol{\lambda}_i(j)$. Then, the triple $(\mathbf{G}_i,\mathbf{W}_1,\mathbf{W}_2)$ is feasible for (P5) due to

$$\begin{aligned}\begin{pmatrix}\mathbf{W}_1 & \mathbf{G}_i\\ \mathbf{G}_i^T & \mathbf{W}_2\end{pmatrix}&=\begin{pmatrix}\boldsymbol{\lambda}_i\boldsymbol{\lambda}_i^T+\frac{\gamma_3}{\gamma_1}\,\mathrm{diag}(\boldsymbol{\lambda}_i) & \boldsymbol{\lambda}_i\mathbf{b}_i^T\\ \mathbf{b}_i\boldsymbol{\lambda}_i^T & \mathbf{b}_i\mathbf{b}_i^T+\frac{\gamma_2}{\gamma_1}\,\mathrm{diag}(\mathbf{b}_i)\end{pmatrix}\\ &=\begin{pmatrix}\boldsymbol{\lambda}_i\\ \mathbf{b}_i\end{pmatrix}\begin{pmatrix}\boldsymbol{\lambda}_i\\ \mathbf{b}_i\end{pmatrix}^T+\frac{1}{\gamma_1}\begin{pmatrix}\gamma_3\,\mathrm{diag}(\boldsymbol{\lambda}_i) & 0\\ 0 & \gamma_2\,\mathrm{diag}(\mathbf{b}_i)\end{pmatrix}\succeq 0.\end{aligned} \tag{27}$$

In addition, we have

$$\gamma_1\kappa(\mathbf{W}_1,\mathbf{W}_2)=\gamma_1\Big(\|\boldsymbol{\lambda}_i\|_2^2+\|\mathbf{b}_i\|_2^2\Big)+\gamma_2\|\mathbf{b}_i\|_1+\gamma_3\|\boldsymbol{\lambda}_i\|_1,$$

and therefore the same objective function for (P3) and (P5) are obtained. This proves any feasible solution of (P5) yields an inner bound for (P3).

We next establish that the proposed inner bound is always equal to (P3) upon satisfying the condition introduced in Theorem 1 and conclude the two problems are equivalent. The equivalence ensures that the stationary point of (P3) (which exhibits Theorem 1 condition) is in fact globally optimal. To show this, the Lagrangian is first formed as

$$\begin{aligned}\mathcal{L}(\mathbf{G}_i,\mathbf{a}_i,\mathbf{W}_1,\mathbf{W}_2,\mathbf{M})=&\sum_{l=1}^{L}\big(\mathbf{p}_i^T(t_l)\mathbf{a}_i-\mathbf{u}_i^T(t_l)\mathbf{G}_i\mathbf{p}_i(t_l)\big)^2\\ &+\gamma_1\kappa(\mathbf{W}_1,\mathbf{W}_2)-\langle\mathbf{M},\mathbf{W}\rangle,\end{aligned}$$

and $\mathbf{M}$ indicates the dual variable associated with the constraint $\mathbf{W}\succeq 0$. In accordance with the block structure of $\mathbf{W}$ in (P5), we define $M_1:=[M]_{11}$, $M_2:=[M]_{12}$, $M_3:=[M]_{22}$, and $M_4:=[M]_{21}$. The optimal solution of (P5) must

(i) null the sub-gradients

$$\begin{aligned}\nabla_{\mathbf{a}_i}\mathcal{L}(\mathbf{G}_i,\mathbf{a}_i,\mathbf{W}_1,\mathbf{W}_2,\mathbf{M})=2\sum_{l=1}^{L}&(\mathbf{p}_i^T(t_l)\mathbf{a}_i\\ &-\mathbf{u}_i^T(t_l)\mathbf{G}_i\mathbf{p}_i(t_l))\mathbf{p}_i(t_l)\end{aligned} \tag{28}$$

$$\begin{aligned}\nabla_{\mathbf{G}_i}\mathcal{L}(\mathbf{G}_i,\mathbf{a}_i,\mathbf{W}_1,\mathbf{W}_2,\mathbf{M})=-2\sum_{l=1}^{L}&(\mathbf{p}_i^T(t_l)\mathbf{a}_i\\ &-\mathbf{u}_i^T(t_l)\mathbf{G}_i\mathbf{p}_i(t_l))\mathbf{u}_i(t_l)\mathbf{p}_i^T(t_l)\\ &-\mathbf{M}_2-\mathbf{M}_4^T\end{aligned} \tag{29}$$

$$\nabla_{\mathbf{W}_1}\mathcal{L}(\mathbf{G}_i,\mathbf{a}_i,\mathbf{W}_1,\mathbf{W}_2,\mathbf{M})=\frac{\gamma_1}{2}\mathbf{I}-\mathbf{M}_1 \tag{30}$$

$$\nabla_{\mathbf{W}_2}\mathcal{L}(\mathbf{G}_i,\mathbf{a}_i,\mathbf{W}_1,\mathbf{W}_2,\mathbf{M})=\frac{\gamma_1}{2}\mathbf{I}-\mathbf{M}_3, \tag{31}$$

and also satisfy

(ii) the complementary slackness condition $\langle\mathbf{M},\mathbf{W}\rangle=0$;

(iii) primal feasibility $\mathbf{W}\succeq 0$;

(iv) dual feasibility $\mathbf{M}\succeq 0$.

Consider the stationary points of (P3), and choose the candidate primal variables $\tilde{\mathbf{a}}_i:=\bar{\mathbf{a}}_i$, $\tilde{\mathbf{G}}_i:=\bar{\boldsymbol{\lambda}}_i\bar{\mathbf{b}}_i^T$, $\tilde{\mathbf{W}}_1:=\bar{\boldsymbol{\lambda}}_i\bar{\boldsymbol{\lambda}}_i^T+\frac{\gamma_3}{\gamma_1}\mathrm{diag}(\bar{\boldsymbol{\lambda}}_i)$, $\tilde{\mathbf{W}}_2:=\bar{\mathbf{b}}_i\bar{\mathbf{b}}_i^T+\frac{\gamma_2}{\gamma_1}\mathrm{diag}(\bar{\mathbf{b}}_i)$; and the dual variables $\tilde{\mathbf{M}}_1:=\frac{\gamma_1}{2}\mathbf{I}$, $\tilde{\mathbf{M}}_3:=\frac{\gamma_1}{2}\mathbf{I}$, $\tilde{\mathbf{M}}_2:=-\sum_{l=1}^{L}\Omega_l(\bar{\mathbf{a}}_i,\bar{\mathbf{b}}_i,\bar{\boldsymbol{\lambda}}_i)\,\mathbf{u}_i(t_l)\mathbf{p}_i^T(t_l)$, and $\tilde{\mathbf{M}}_4:=\tilde{\mathbf{M}}_2^T$. Then, condition (i) holds because the sub-gradients (28)-(31) are zero when substituting the introduced primal and dual variables. The requirement (ii) is also true since

$$\begin{aligned}\langle\tilde{\mathbf{M}},\tilde{\mathbf{W}}\rangle&=\langle\tilde{\mathbf{M}}_1,\tilde{\mathbf{W}}_1\rangle+\langle\tilde{\mathbf{M}}_3,\tilde{\mathbf{W}}_2\rangle+2\langle\tilde{\mathbf{M}}_2,\tilde{\mathbf{G}}_i\rangle\\ &=\frac{\gamma_1}{2}\mathrm{Tr}\Big(\bar{\boldsymbol{\lambda}}_i\bar{\boldsymbol{\lambda}}_i^T+\frac{\gamma_3}{\gamma_1}\,\mathrm{diag}(\bar{\boldsymbol{\lambda}}_i)\Big)\\ &\quad+\frac{\gamma_1}{2}\mathrm{Tr}\Big(\bar{\mathbf{b}}_i\bar{\mathbf{b}}_i^T+\frac{\gamma_2}{\gamma_1}\,\mathrm{diag}(\bar{\mathbf{b}}_i)\Big)\\ &\quad-2\mathrm{Tr}\Bigg(\sum_{l=1}^{L}\Omega_l(\bar{\mathbf{a}}_i,\bar{\mathbf{b}}_i,\bar{\boldsymbol{\lambda}}_i)\mathbf{p}_i^T(t_l)\bar{\mathbf{b}}_i\mathbf{u}_i^T(t_l)\bar{\boldsymbol{\lambda}}_i\Bigg)\\ &=\frac{1}{2}\mathrm{Tr}\big(\gamma_1\bar{\boldsymbol{\lambda}}_i\bar{\boldsymbol{\lambda}}_i^T+\gamma_3\,\mathrm{diag}(\bar{\boldsymbol{\lambda}}_i)\big)\\ &\quad+\frac{1}{2}\mathrm{Tr}\big(\gamma_1\bar{\mathbf{b}}_i\bar{\mathbf{b}}_i^T+\gamma_2\,\mathrm{diag}(\bar{\mathbf{b}}_i)\big)\\ &\quad-\mathrm{Tr}\big(\gamma_1\bar{\boldsymbol{\lambda}}_i\bar{\boldsymbol{\lambda}}_i^T+\gamma_3\,\mathrm{diag}(\bar{\boldsymbol{\lambda}}_i)\big)=0,\end{aligned}$$

where the last equality follows from (25). Moreover, (iii) is confirmed similar to (27). In order to meet the last criterion (iv), according to a Schur complement argument [40], it is sufficient to invoke $\|\tilde{\mathbf{M}}_2\|\leq\gamma_1/2$.

Consequently, by choosing the proposed candidates that have been proved to be optimal, one can easily verify (P5) coincides with (P3). This completes the proof.

Acknowledgments

The research was supported in part by the US National Science Foundation (NSF) award No. 1523374. The first two authors contributed equally to this work.

References

[1] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," *Nature Genetics*, vol. 34, no. 2, pp. 166–176, 2003.

[2] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. Di Bernardo, "How to infer gene networks from expression profiles," *Molecular Syst. Biol.*, vol. 3, no. 1, p. 78, 2007.

[3] M. Hecker, S. Lambeck, S. Toepfer, E. Van Someren, and R. Guthke, "Gene regulatory network inference: Data integration in dynamic models—A review," *Biosystems*, vol. 96, no. 1, pp. 86–103, 2009.

[4] W.-P. Lee and W.-S. Tzou, "Computational methods for discovering gene networks from expression data," *Briefings Bioinformat.*, vol. 10, no. 4, pp. 408–423, 2009.

[5] R. De Smet and K. Marchal, "Advantages and limitations of current network inference methods," *Nature Rev. Microbiol.*, vol. 8, no. 10, pp. 717–729, 2010.

[6] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky, "Revealing strengths and weaknesses of methods for gene network inference," *Proc. Nat. Acad. Sci. USA*, vol. 107, no. 14, pp. 6286–6291, 2010.

[7] B. Jia and X. Wang, "Gene regulatory network inference by point-based gaussian approximation filters incorporating the prior information," *EURASIP J. Bioinformat. Syst. Biol.*, vol. 2013, no. 1, pp. 1–16, 2013.

[8] R. C. Jansen, "Studying complex biological systems using multifactorial perturbation," *Nature Rev. Genetics*, vol. 4, no. 2, pp. 145–151, 2003.

[9] T. E. Ideker, V. Thorsson, and R. M. Karp, "Discovery of regulatory interactions through perturbation: Inference and experimental design," in *Proc. Pacific Symp. Biocomput.*, vol. 5, pp. 302–313, 2000.

[10] J. Tegner, M. S. Yeung, J. Hasty, and J. J. Collins, "Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 10, pp. 5944–5949, 2003.

[11] 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*, vol. 301, no. 5629, pp. 102–105, 2003.

[12] M. Bansal, G. Della Gatta, and D. Di Bernardo, "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles," *Bioinformatics*, vol. 22, no. 7, pp. 815–822, 2006.

[13] B. A. Logsdon and J. Mezey, "Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations," *PLOS Comput. Biol.*, vol. 6, no. 12, pp. e1001014, 2010.

[14] G. Krouk, J. Lingeman, A. M. Colon, G. Coruzzi, D. Shasha, et al., "Gene regulatory networks in plants: Learning causality from time and perturbation," *Genome Biol.*, vol. 14, no. 6, p. 123, 2013.

[15] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLOS Comput. Biol.*, vol. 9, no. 5, pp. e1003068, 2013.

[16] H. Bolouri, "Modeling genomic regulatory networks with big data," *Trends Genetics*, vol. 30, no. 5, pp. 182–191, 2014.

[17] B. C. Goodwin, "Oscillatory behavior in enzymatic control processes," *Adv. Enzyme Regulation*, vol. 3, pp. 425–437, 1965.

[18] T. Chen, H. L. He, G. M. Church, et al., "Modeling gene expression with differential equations," in *Proc. Pacific Symp. Biocomput.*, vol. 4, p. 4, 1999.

[19] A. Ay and D. N. Arnosti, "Mathematical modeling of gene expression: A guide for the perplexed biologist," *Critical Rev. Biochemistry Molecular Biol.*, vol. 46, no. 2, pp. 137–151, 2011.

[20] L. Bintu, N. E. Buchler, H. G. Garcia, U. Gerland, T. Hwa, J. Kondev, and R. Phillips, "Transcriptional regulation by the numbers: Models," *Current Opinion Genetics Develop.*, vol. 15, no. 2, pp. 116–124, 2005.

[21] L. Bintu, N. E. Buchler, H. G. Garcia, U. Gerland, T. Hwa, J. Kondev, T. Kuhlman, and R. Phillips, "Transcriptional regulation by the numbers: Applications," *Current Opinion Genetics Develop.*, vol. 15, no. 2, pp. 125–135, 2005.

[22] A. Meister, Y. H. Li, B. Choi, W. H. Wong, et al., "Learning a nonlinear dynamical system model of gene regulation: A perturbed steady-state approach," *Ann. Appl. Statist.*, vol. 7, no. 3, pp. 1311–1333, 2013.

[23] R. Khanin and V. Vinciotti, "Computational modeling of post-transcriptional gene regulation by microRNAs," *J. Comput. Biol.*, vol. 15, no. 3, pp. 305–316, 2008.

[24] J. Hausser, A. P. Syed, N. Selevsek, E. van Nimwegen, L. Jaskiewicz, R. Aebersold, and M. Zavolan, "Timescales and bottlenecks in miRNA-dependent gene regulation," *Molecular Syst. Biol.*, vol. 9, no. 1, p. 711, 2013.

[25] E. van Rooij, "The art of microRNA research," *Circulation Res.*, vol. 108, no. 2, pp. 219–234, 2011.

[26] G. Hutvágner and P. D. Zamore, "A microRNA in a multiple-turnover RNAi enzyme complex," *Science*, vol. 297, no. 5589, pp. 2056–2060, 2002.

[27] J. G. Doench and P. A. Sharp, "Specificity of microrna target selection in translational repression," *Genes Develop.*, vol. 18, no. 5, pp. 504–511, 2004.

[28] C. S. Möller-Levet, K.-H. Cho, H. Yin, and O. Wolkenhauer, "Modelling gene expression time-series with radial basis function neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2004, vol. 2, pp. 1191–1195.

[29] Y. Luan and H. Li, "Clustering of time-course gene expression data using a mixed-effects model with B-splines," *Bioinformatics*, vol. 19, no. 4, pp. 474–482, 2003.

[30] S. Wichert, K. Fokianos, and K. Strimmer, "Identifying periodically expressed transcripts in microarray time series data," *Bioinformatics*, vol. 20, no. 1, pp. 5–20, 2004.

[31] Y. Luan and H. Li, "Model-based methods for identifying periodically expressed genes based on time course microarray gene expression data," *Bioinformatics*, vol. 20, no. 3, pp. 332–339, 2004.

[32] M. Rooman, J. Albert, Y. Dehouck, and A. Haye, "Detection of perturbation phases and developmental stages in organisms from dna microarray time series data," *PlOS ONE*, vol. 6, no. 12, pp. e27948, 2011.

[33] E. F. Glynn, J. Chen, and A. R. Mushegian, "Detecting periodic patterns in unevenly spaced gene expression time series using Lomb–Scargle periodograms," *Bioinformatics*, vol. 22, no. 3, pp. 310–316, 2006.

[34] P. Chaudhuri and J. S. Marron, "Sizer for exploration of structures in curves," *J. Am. Statistical Assoc.*, vol. 94, no. 447, pp. 807–823, 1999.

[35] S. Zhang, Q. Li, J. Liu, and X. J. Zhou, "A novel computational framework for simultaneous integration of multiple types of genomic data to identify microRNA-gene regulatory modules," *Bioinformatics*, vol. 27, no. 13, pp. i401–i409, 2011.

[36] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statistical Soc.: Series B (Statistical Methodol.)*, vol. 67, no. 2, pp. 301–320, 2005.

[37] T. Shimamura, S. Imoto, R. Yamaguchi, A. Fujita, M. Nagasaki, and S. Miyano, "Recursive regularization for inferring gene networks from time-course gene expression profiles," *BMC Syst. Biol.*, vol. 3, no. 1, p. 41, 2009.

[38] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.

[39] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[40] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[41] M. Mardani, G. Mateos, and G. Giannakis, "Decentralized sparsity-regularized rank minimization: Algorithms and applications," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5374–5388, Nov. 2013.

[42] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: A survey and extensions," *Math. Methods Operations Res.*, vol. 66, no. 3, pp. 373–407, 2007.

[43] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," *Nature Rev. Molecular Cell Biol.*, vol. 9, no. 10, pp. 770–780, 2008.

[44] J. Bähler, "Cell-cycle control of gene expression in budding and fission yeast," *Annu. Rev. Genet.*, vol. 39, pp. 69–94, 2005.

[45] A. Greenfield, A. Madar, H. Ostrer, and R. Bonneau, "Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models," *PlOS ONE*, vol. 5, no. 10, p. e13397, 2010.

[46] C. A. Penfold and D. L. Wild, "How to infer gene networks from expression profiles, revisited," *Interface Focus*, vol. 1, no. 6, pp. 857–870, 2011.

[47] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Molecular Biol. Cell*, vol. 9, no. 12, pp. 3273–3297, 1998.

[48] M. Breker and M. Schuldiner, "The emergence of proteome-wide technologies: Systematic analysis of proteins comes of age," *Nature Rev. Molecular Cell Biol.*, vol. 15, no. 7, pp. 453–464, 2014.

[49] M. Rabani, J. Z. Levin, L. Fan, X. Adiconis, R. Raychowdhury, M. Garber, A. Gnirke, C. Nusbaum, N. Hacohen, N. Friedman, et al., "Metabolic labeling of RNA uncovers principles of RNA production and degradation dynamics in mammalian cells," *Nature Biotechnol.*, vol. 29, no. 5, pp. 436–442, 2011.

[50] T. Maier, A. Schmidt, M. Güell, S. Kühner, A.-C. Gavin, R. Aebersold, and L. Serrano, "Quantification of mRNA and protein and integration with protein turnover in a bacterium," *Molecular Syst. Biol.*, vol. 7, no. 1, p. 511, 2011.

[51] M. K. Doherty and R. J. Beynon, "Protein turnover on the scale of the proteome," *Expert Rev Proteomics*, vol. 3, no. 1, pp. 97–110, 2006.

[52] J. M. Peña, "B-splines and optimal stability," *Math. Comput.*, vol. 66, no. 220, pp. 1555–1560, 1997.

[53] X. Hong, S. Chen, Y. Gong, and C. Harris, "Nonlinear equalization of Hammerstein OFDM systems," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5629–5639, Nov. 1, 2014.

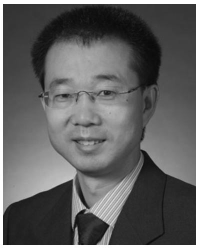
[54] C. De Boor, et al., *A Practical Guide to Splines*. Berlin, Germany: Springer, 1978.

[55] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.

Mahdi Zamanighomi received the BSc degree in electrical engineering from Sharif University of Technology in 2012, and the PhD degree in electrical and computer engineering from Iowa State University, 2015. Since May 2015, he has been a postdoctoral fellow in the Wong Lab at Stanford University. His research interests include genomics and bioinformatics, systems biology, high-dimensional statistical inference, and signal processing.

Mostafa Zamanian received the BS degree in biochemistry and the PhD degree in neuroscience from Iowa State University. His doctoral work was concerned with infectious disease genomics and drug target discovery. Upon graduation, he migrated to a joint postdoctoral appointment at McGill University. He is currently a research fellow in the Andersen Laboratory at Northwestern University. His research involves the application of both computational and wet lab approaches to the study of "Neglected Diseases." He currently works at the intersection of quantitative genetics, genome editing, and drug resistance.

Michael Kimber received the BSc (Hons.) degree in 1998 and the PhD degree in molecular parasitology in 2001 from Queen's University Belfast, United Kingdom. He is currently an associate professor of physiology in the Department of Biomedical Sciences, Iowa State University's College of Veterinary Medicine. His broad background is in helminth biology and, in particular, parasitic nematodes. His research program integrates a diverse range of molecular biology and physiology techniques to better understanding the biology of parasitic nematode systems and processes. A recent focus is the use of next-gen deep-sequencing strategies to study small RNA species called microRNAs (miRNA) as effectors of gene manipulation. An author of over 30 publications, he also serves as a member of multiple NIH Study Section panels and international granting bodies.



Zhengdao Wang (S'00-M'02-SM'08) received the BS degree in electronic engineering and information science from the University of Science and Technology of China (USTC), 1996, the MSc degree in electrical and computer engineering from the University of Virginia, 1999, and the PhD degree in electrical and computer engineering from the University of Minnesota, 2002. He is currently with the Department of Electrical and Computer Engineering, Iowa State University. His interests are in the areas of signal processing, communications, and information theory. He served as an associate editor for the *IEEE Transactions on Vehicular Technology* from April 2004 to April 2006, an associate editor for *IEEE Signal Processing Letters* between August 2005 and August 2008, and an associate editor for the *IEEE Transactions on Signal Processing* between 2013 and 2015. He co-received the IEEE Signal Processing Magazine Best Paper Award in 2003 and the IEEE Communications Society Marconi Paper Prize Award in 2004, and the *EURASIP Journal on Advances in Signal Processing* Best Paper Award, in 2009. He is serving as an editor for The IEEE Signal Processing Society Online Video Library, and an associate editor for the *IEEE Transactions on Wireless Communications*. He is a senior member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.