Gene regulatory network inference from perturbed time-series expression data via ordered dynamical expansion of non-steady state actors

Mahdi Zamanigomi ⋆, Mostafa Zamanian †‡, Michael Kimber †, Zhengdao Wang ⋆§

⋆ Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 50010 USA
† Department of Molecular Biosciences, Northwestern University, Evanston, IL, USA
‡ Department of Biomedical Sciences, Iowa State University, Ames, IA, 50010 USA
§Authors contributed equally to this manuscript, E-mail: zhengdao@iastate.edu

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 often difficult to scale, 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, present 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.

I. 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]. Of particular interest is the goal of gene network inference using perturbed gene expression data [7], [8], [9], [10], [11], [12], [13], [14], [15], 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.

Notation: Throughout this paper \( \{d, i, j, k, l\} \) counts integer numbers. Column vectors and matrices are indicated by bold lower-case letters. The paper is concerned with datasets with known points of 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.

\[
X^p := \begin{pmatrix}
\ldots & x_1(t_0) & x_1(t_1) & \ldots & x_1(t_L) \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\ldots & x_m(t_0) & x_m(t_1) & \ldots & x_m(t_L)
\end{pmatrix}
\]
B. 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 \(dy_j(t)/dt\) 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 \(|dx_i(t)/dt| \gg 0\), 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, \(\Delta x_i\), is coupled to a delayed change in protein expression, \(\Delta y_j\). This shift in protein availability leads, through the immediate regulatory network of the perturbed protein, to changes in the expression levels of other genes.

Consider the set of all genes that are affected by \(\Delta y_j\) 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 \(X_G(t) := \{x_i(t)|i \in G(t)\}\) and \(X_M(t) := \{x_i(t)|i \in M(t)\}\), respectively. We further define the collection of protein expression levels for subset \(G\) as \(Y_G(t) := \{y_i(t)|i \in G(t)\}\).

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.

C. Governing regulatory equations

Gene and protein expression dynamics are often modeled in the form of ordinary differential equations \([16], [17], [18]\), 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 Figure 1.

This circuit can be represented in the following form:

\[
\frac{dx_i(t)}{dt} = \tau_i f_i(Y_G(t)) - (\lambda_i^{RNA} + g_i(X_M(t))) x_i(t) \quad (1)
\]

\[
\frac{dy_j(t)}{dt} = (r_j - h_j(X_M(t))) x_i(t) - \lambda_j^{prot} y_j(t), \quad (2)
\]

where \(\tau_i\) is the rate of transcription when RNA polymerase (RNAP) is bound, \(f_i(Y_G(t))\) is the probability of RNAP binding, \(\lambda_i^{RNA}\) is the rate of basal RNA degradation, \(g_i(X_M(t))\) incorporates the effect of miRNA-mediated RNA degradation, \(r_j\) is the rate of translation, \(h_j(X_M(t))\) accounts for the effect of miRNA-mediated translational inhibition, and \(\lambda_j^{prot}\) is the rate of protein degradation. It follows from the biological definitions of the system that parameters \(\tau_i\), \(\lambda_i^{RNA}\), \(r_j\), and \(\lambda_j^{prot}\) are to be positive and \(h_j(X_M(t)) \leq r_j\).

D. Protein-mediated regulation

For each gene, \(i\), we employ an existing statistical thermodynamic framework \([19], [20]\) to model the equilibrium probability of RNAP binding to a gene of interest as a function of protein regulators, \(f_i(Y_G(t))\). We extend a previous derivation of multiple protein regulators operating on a single gene \([21]\) and explicitly show that the general form can be expressed as a function of non-steady state genes, \(G(t)\) (cf. 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.

\[
f_i(Y_G(t)) = \frac{\alpha_{i0} + \sum_{j=1}^{N(t)} \alpha_{ij} \prod_{k \in S_i(t)} y_k(t)}{1 + \sum_{j=1}^{N(t)} b_{ij} \prod_{k \in S_i(t)} y_k(t) + \prod_{k \in S_i(t)} y_k(t)} \quad (3)
\]

where \(S_i(t), 0 \leq j \leq N(t)\), is the list of all possible protein products of genes within set \(G(t)\) that interact to form regulatory complexes. For instance when \(G(t) = \{1, 2\}\), there are \(N(t)+1 = 4\) complexes as the empty set \(S_{i0} = \{0\}\), \(S_{i1} = \{1\}\), \(S_{i2} = \{2\}\), and \(S_{i3} = \{1, 2\}\). To reduce the complexity of this model, we restrict \(S_i(t)\) 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. The coefficients \(0 \leq \alpha_{ij} \leq b_{ij}\) 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 \(b_{i0} = 1\).

E. miRNA-mediated regulation

To account for the effects of miRNA regulation on each gene, we draw on previous mass-law (linear) models \([22], [23]\) that acknowledge two primary routes of inhibitory regulation: (i) cleavage or degradation of target transcript and (ii) translational repression. These are represented by functions \(g_i(X_M(t))\) and \(h_i(X_M(t))\), respectively. The former is a modifier of the RNA degradation rate...
and inference steps, although designed for a normalized gene expression dataset involving a precise perturbation, is robust and flexible.

A. 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

\[
x_i(t) = \sum_{d=1}^{D} \theta_{i,d} \varphi_{i,d}(t) = \varphi(t)^T \theta_i,
\]

where \( D \) is the total number of bases and \( \theta_{i,d} \) the coefficient of the \( d \)-th basis function, \( \varphi_{i,d}(t) \). The basis functions are chosen to take the form of a B-spline (cf. 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.

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

\[
(P1) \quad \min_{\theta_i} \left\| \begin{array}{c} \sum_{j=1}^{L} \left( x_i(t_j) - \varphi(t_j)^T \theta_i \right) \end{array} \right\|_2^2 + \gamma_0 \theta_i^T K \theta_i,
\]

where the roughness penalty \( \theta_i^T K \theta_i = \int_{t_1}^{t_L} (d^2 x_i(t)/dt^2)^2 \, dt \) and \( K \) is a roughness matrix with the \((j,k)\)-th entry \( \int_{t_j}^{t_k} \varphi_{i,j}(t) \varphi_{i,k}(t) \, dt \). Here, the first term is intended to diminish noise within measurements and the second term is intended to smooth our approximations. The parameter \( \gamma_0 \) 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

\[
\frac{d\hat{x}_i(t)}{dt} \approx \frac{\hat{x}_i(t + \Delta t) - \hat{x}_i(t)}{\Delta t}.
\]

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.

B. 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 [27], [28], 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 \([t_1, t_L], (r+1) t_{1,L} \) 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 \( R \geq D \).

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 \( t > r t_{1,L} \) 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

\[
\text{Algorithm 1}
\]

\begin{align}
g_t(X_{M(t)}) &= \sum_{j \in X_{M(t)}} \lambda^{RNA}_{ij} x_j(t) \\
h_t(X_{M(t)}) &= \sum_{j \in X_{M(t)}} \lambda^{Prot}_{ij} x_j(t)
\end{align}

where both \( \lambda^{RNA}_{ij} \) and \( \lambda^{Prot}_{ij} \) are greater than or equal to zero.

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 [24], [25], [26]. This constraint takes the following mathematical form

\[
\mathbb{I}_R + \{ \lambda^{RNA}_{ij} \} + \mathbb{I}_R + \{ \lambda^{Prot}_{ij} \} = 1.
\]

III. NETWORK INFERENCE ALGORITHM

Sub-sections III-A - III-D 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 Figure 2. This linear ordering of post-processing

\[
\text{Fig. 2. Overview of gene inference pipeline, beginning with a normalized gene expression dataset. The first stage involves the estimation of all gene trajectories as noise-free and continuous curves (P1), followed by segmentation into equally-spaced intervals for detection of significant changes in expression. The time-dependent expansion of } G(t) \text{ and } M(t) \text{, along with the result of (P1), seed downstream network inference. In the next stage, (P2) is used to estimate protein expression, and finally all obtained results are considered (P1), seed downstream network inference. In the next stage, (P2) is used to estimate protein expression, and finally all obtained results are considered.}
\]

constant, \( \lambda^{RNA} \), while the latter detacks from RNA available to the translational machinery without affecting RNA concentration as assayed. These functions are shown below.

\[
g_t(X_{M(t)}) = \sum_{j \in X_{M(t)}} \lambda^{RNA}_{ij} x_j(t) \\
h_t(X_{M(t)}) = \sum_{j \in X_{M(t)}} \lambda^{Prot}_{ij} x_j(t)
\]

Before the translational machinery without affecting RNA concentration as constant,
Consider gene expressions at times $t_l$, $1 \leq l \leq L$. Setting all available gene expressions in equation (9), we arrive at

$$A_i \left( -r_i, z_i^T, \alpha_i^T \right)^T = 0,$$

where

$$A_i := \begin{array}{c}
\begin{array}{c}
x_i(t_1) \\
x_i(t_2) \\
\vdots \\
x_i(t_L)
\end{array} \\
\begin{array}{c}
x_M(t_1)^T \\
x_M(t_2)^T \\
\vdots \\
x_M(t_L)^T
\end{array} \\
\begin{array}{c}
b_i(t_1)^T \\
b_i(t_2)^T \\
\vdots \\
b_i(t_L)^T
\end{array}
\end{array},$$

$0(t_l)$ is the zero column vector with length $\text{card}(M(t_l))$, $\lambda_i^{\text{prot}}$, and $\alpha_i^T$ are in turn used to approximate unknown parameters. However the sparsity in $z_i^T$, given that only a small number of miRNAs typically act on a common gene [32], reduces the number of required equations.

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

$$(P2) \quad \min_{(z_i, \alpha_i)} \left\| A_i \left( -1, z_i \right)^T \right\|_2 + \gamma_z \|z_i\|_1 + \gamma_\alpha \alpha_i^T K \alpha_i,$$

subject to $z_i \geq 0$ and $x_M(t_L)^T \cdot 0(t_l)^T z_i \leq x_i(t_l) \quad \forall 1 \leq l \leq L$

where $\gamma_z$ and $\gamma_\alpha$ are chosen using cross validation. The second constraint ensures that the total rate of translation, $r_i - h_i(\lambda_i^{\text{prot}})$, 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, $\lambda_i^{\text{prot}}$. In the absence of this experimental data, we can fix the value of $\lambda_i^{\text{prot}}$ to 1 for the entire system and still achieve accurate network reconstruction as shown in subsequent sections.

D. Gene regulatory 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 $\tau_i$. 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 a large number of genes and gene products under consideration. Our goal is to estimate these coefficients so that the ODE models can be temporally fitted to large 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.

[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. If more complex change detection schemes are preferred, a number of alternative approaches can be adapted for this purpose [29, 30, 31].

C. Modeling and estimation of protein expression

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

$$y_i(t) = \sum_{d=1}^{D} \alpha_{i,d} \varphi_d(t) = \varphi(t)^T \alpha_i,$$ (8)

Our objective is first to find $\alpha_i$ 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

$$y_i'(t) = \varphi'(t)^T \alpha_i,$$

and ODE (2) is consequently represented as

$$\varphi'(t)^T \alpha_i = \left( r_i - \sum_{j \in M(t)} \lambda_j^{\text{prot}} x_j(t) \right) x_i(t) - \lambda_i^{\text{prot}} \varphi'(t)^T \alpha_i.$$ (9)

We rewrite the above equation in the following form

$$x_i(t) r_i - x_M(t)^T \lambda_i^R - b_i(t)^T \alpha_i = 0,$$

where $b_i(t)^T := [\lambda_i^{\text{prot}} \varphi'(t)^T + \varphi'(t)^T]^T$ and $\lambda_i^R(t)$ is the column vector with entries $\lambda_j^{\text{prot}}$, $\forall j \in M(t)$. The mRNA expressions corresponding to $\lambda_i^R(t)$ are indicated by the vector $x_M(t)$ such that both vectors, $\lambda_i^R(t)$ and $x_M(t)$, have the same index order. For notational convenience, we assume that all entries of $x_M(t)$ are multiplied by $x_i(t)$.
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., $dx_{i}(t)/dt = dy_{i}(t)/dt \approx 0$). As previously mentioned, the perturbation of protein-encoding gene $x_i^0(t_1)$ 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 $t_1$, 1 $\leq$ $l$ $\leq$ $L$, with the exception of $x_i^0(t_1)$, the term $\tau_i f_i(y_i(t))$ in equation (1) can be rewritten as follows

$$\tau_i f_i(y_i(t)) = \frac{\tau_i a_{ij} + \sum_{j=1}^{N(t)} \tau_i a_{ij} \prod_{k \in S_{ij}(t)} y_k(t_1)}{1 + \sum_{j=1}^{N(t)} b_{ij} \prod_{k \in S_{ij}(t)} y_k(t_1)} = p_i^T(t_1)a_i, \quad p_i^T(t_1)b_i,$$

(11)

where $a_i$ is a vector with $(j+1)$th entry $\tau_i a_{ij}, 0 \leq j \leq N(t_i)$. The $(j+1)$th element of vector $p_i(t_1)$ is described by $\prod_{k \in S_{ij}(t_1)} y_k(t_1)$ when $0 \leq j \leq N(t_i)$ and zero for $N(t_i)+1 \leq j \leq N(t_i)$. Vector $b_i$ is defined such that the first entry is 1 and $(j+1)$th, $1 \leq j \leq N(t_i)$, is $b_j$.

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

We also represent

$$\left( \lambda^{RNA}_i + \sum_{j \in M(t_i)} \lambda^{RNA}_{ij} x_j(t_1) \right) x_i(t_1) + \frac{dx_i(t)}{dt}|_{t=t_1} = u_i^T(t_1)\lambda_i,$$

(12)

in which $u_i(t_1)$ and $\lambda_i$ are defined as follows. First and second entries of vector $u_i(t_1)$ are $dx_i(t)/dt|_{t=t_1}$ and $x_i(t_1)$, respectively. The remaining entries are $x_j(t_1)x_j(t_1)$, $j \in M(t_i)$. Making the same arrangement of array as $u_i(t_1)$, vector $\lambda_i$ is determined by first entry 1, second entry $\lambda^{RNA}_i$, and subsequent entries $\lambda^{RNA}_{ij}, j \in M(t_i)$.

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

$$\Omega_i(a_i, b_i, \lambda_i) := p_i^T(t_1)a_i - u_i^T(t_1)\lambda_i b_i, \quad p_i(t_1) = 0.$$

(13)

**Algorithm:** We need to solve the non-convex problem

$$\min_{\{a_i, b_i, \lambda_i\}} \Gamma(a_i, b_i, \lambda_i),$$

subject to

$$0 \leq a_i \leq b_i, 0 \leq \lambda_i, \quad b_i(1) = 1, \lambda_i(1) = 1, \lambda_i(2) = \lambda^{RNA}_i,$$

(14)

with

$$\Gamma(a_i, b_i, \lambda_i) := \sum_{i=1}^{L} \Omega_i(a_i, b_i, \lambda_i)^2 + \gamma_1 \left( \|\lambda_i\|_2^2 + \|b_i\|_2^2 \right) + \gamma_2 \left( \|a_i\|_1 + \|b_i\|_1 \right) + \gamma_3 \|\lambda_i\|_1,$$

The first term in the above equation follows from (13). The second term associated with $\gamma_1/2$ ensures that neither $b_i$ nor $\lambda_i$ tend to infinity in the 2-norm sense. Due to the assumption that each gene has only a few regulators, 1-norm regularizations are considered to encourage sparse solutions. Note that in the absence of miRNAs (all $\lambda_{ij}^{RNA} = 0$), the terms associated with $\gamma_1$ and $\gamma_3$ 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 $\{a_i, b_i\}$ for fixed $\lambda_i$ 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 [33], cf. Appendix C. Here, given the absence of dual variables, ADMM is reduced to simple alternating minimization. The proposed solver entails an iterative procedure comprising two steps per iteration $k = 1, 2, \ldots$

**Algorithm 1:** Gene regulatory inference

```plaintext
input a_i, b_i, \lambda_i
initialize a_i[0], b_i[0], and \lambda_i[0] at random with respect to
b_i(1) = 1, \lambda_i(1) = 1, and \lambda_i(2) = \lambda_i^{RNA}.
for k = 0, 1, \ldots do
[S1] Update primal variables a_i and b_i:
\{a_i[k]+1, b_i[k]+1\} = arg \min_{(a_i, b_i)} \Gamma(a_i, b_i, \lambda_i[k])
subject to 0 \leq a_i \leq b_i
b_i(1) = 1.
[S2] Update primal variable \lambda_i:
\lambda_i[k+1] = arg \min \Gamma(a_i[k], b_i[k], \lambda_i)
subject to \lambda_i \geq 0
\lambda_i(1) = 1, \lambda_i(2) = \lambda_i^{RNA}.
end for
return a_i, b_i, \lambda_i
```

This iterative procedure implements a block coordinate descent method [34]. 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, $\{b_i, a_i\}$ and $\lambda_i$.

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 $\{b_i, a_i, \lambda_i\}$. Motivated by the proposition 1 in [35], the next theorem offers a global optimality certificate upon the convergence of the solver.

**Theorem 1.** Let $\{a_i, b_i, \lambda_i\}$ be a stationary point of (P3). If

$$\left\| \sum_{i=1}^{L} \Omega_i(a_i, b_i, \lambda_i)u_i(t_1)p_i^T(t_1) \right\| \leq \frac{\gamma_1}{2},$$

then $\{a_i, b_i, \lambda_i\}$ is the globally optimal solution of (P3).

**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, Theorem 4.5 and 4.9 in [36], 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, $\lambda_i^{RNA}$, are available. However, experimentally measuring $\lambda_i^{RNA}$ is a difficult task. We offer a simple modification to the
algorithms so that network inference can be still obtained without prior knowledge of RNA degradation rates.

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

\[ \Omega_i(a_i, b_i, c_i) := p_i^T(t_i)(a_i - b_i, \frac{dx_i(t_i)}{dt} + c_i x_i(t_i)) = 0, \quad (16) \]

and \( c_i := \lambda^R_{i, RNA} b_i \). Employing the above reformulation, unknown variables \( a_i, b_i, \) and \( c_i \) are estimated through

(P4) arg \[ \min_{\{a_i, b_i, c_i\}} \sum_{i=1}^{L} \Omega_i(a_i, b_i, c_i) + \gamma_2(\|a_i\|_1 + \|b_i\|_1 + \|c_i\|_1) \]

subject to \( 0 \leq a_i \)

\[ a_i \leq b_i \]

\[ \lambda_{min} b_i \leq c_i \leq \lambda_{max} b_i, \quad (17) \]

where \( \lambda_{min} \) and \( \lambda_{max} \) specify an lower and upper bound for \( \lambda^R_{i, RNA} \), respectively. Variable \( c_i \) is introduced to remove \( \lambda^R_{i, RNA} \) 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 (17) to (P4) to tighten the feasible set of solutions. Given that \( \lambda^R_{i, RNA} / \tau_i \geq 1 \), we can take on the additional constraint \( a_i \leq c_i \). In the subsequent simulations, \( \lambda_{min} \) is in the near-zero range \([0.001, 0.01]\), and \( \lambda_{max} \) 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.

IV. SIMULATIONS

A. 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

\[
\begin{align*}
\frac{dx_1(t)}{dt} &= 0.1 + 0.05y_1(t)y_2(t) + 0.025y_1(t)y_3(t) \\
&\quad + 1 + 0.1y_1(t) + 10y_3(t) + 0.05y_1(t)y_2(t) + 0.025y_1(t)y_3(t) \\
&\quad - 0.1x_1(t), \\
\frac{dx_2(t)}{dt} &= 0.1 + 0.1y_1(t) + 0.1y_1(t)y_2(t) \\
&\quad + 1 + 0.1y_1(t) + 0.1y_1(t)y_2(t) + 10y_1(t)y_3(t) \\
&\quad - 0.1x_2(t), \\
\frac{dx_3(t)}{dt} &= 0.1 + 0.1y_1(t) + 0.1y_1(t)y_2(t) \\
&\quad + 1 + 0.1y_1(t) + 0.1y_1(t)y_2(t) - 0.1x_3(t),
\end{align*}
\]

(18)

and the following system of ODEs for protein expression

\[
\begin{align*}
\frac{dy_1(t)}{dt} &= x_1(t) - 0.5y_1(t), \\
\frac{dy_2(t)}{dt} &= 2x_2(t) - 0.5y_2(t), \\
\frac{dy_3(t)}{dt} &= x_3(t) - 0.5y_3(t).
\end{align*}
\]

(19)

The above toy model, visualized in Figure 4, is provided to better explain our algorithms. Although a small network is examined, many of the same qualitative characteristics of large network are investigated in this example. The explicit system of ODEs, describing the kinetics of the system [37], 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 (18) and (19) 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. Figure 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 Figure 6.

We finally examine Algorithm 1, (P3), for the goal of network recovery. In this scenario, our target is to estimate vectors \( a_i \) and \( b_i \). We assume that the degradation rates are known in advance and therefore, since the system does not contain any miRNA in this particular example, \( \lambda_i \) is completely at hand. Let us consider gene 3 where the true value of \( a_3 = (0.1, 0.1, 0, 0, 0, 0) \) and \( b_3 = (1, 0, 0, 1, 0, 0, 0, 0) \). Vectors \( a_3 \) and \( b_3 \) are indexed with
regard to
\[ p_3(t_i) = \]
\[ (1, y_1(t_i), y_2(t_i), y_3(t_i), y_1(t_i)y_2(t_i), y_1(t_i)y_3(t_i), y_2(t_i)y_3(t_i)). \]

Applying our method, we obtain \( a_3 \approx (0.1, 0, 0.083, 0, 0, 0, 0) \) and \( b_3 \approx (1, 0, 0.083, 0.08, 0, 0, 0) \). Table I 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, \( b_{3,2}(6) = 10 \). The difficulty here is due to the sharp change in \( y_1 \) (Figure 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.

**B. 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 [38]. Each network dataset includes a simulated time series of gene expression in response to five 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 used 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 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 were used to infer gene regulatory networks and to inspect predictions of network topology using the official scoring pipeline. First, we used (P1) to produce smooth and continuous gene expression trajectories from the discrete and noisy time series datasets (Figure 7). Perturbed genes were identified and incorporated as described in Section III-B. Network inference was carried out using Algorithm 1. In the absence of RNA degradation rates, \( \lambda_{\text{min}} \) was set to either 0.001 or 0.01, and \( \lambda_{\text{max}} \) was set to 0.1 or 1. If a directed network edge was identified, the probability of the edge was set to 1 for weighted edges, and 0 otherwise. This was done to allow scoring of our network with the provided scripts, given our non-probabilistic formulation. Algorithm 1 minimization values were filtered against abnormal values that could represent underfitting and overfitting of data.

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 (Supplemental Figure 1). These results compare very favorably to other time series-based methods applied to the same datasets [39]. In fact, for Networks 1 and 2, the AUROC and AUPR values represent improvements over the top reported results.

**C. Large network inference from perturbed yeast time series data**

This section will be completed shortly.
Continuous modeling of gene expressions from noisy time–series measurements

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.

V. 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. 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 good results 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 non-coding RNAs. Technologies are continually being improved for the purpose of capturing these data in a genome-wide manner [40], [41], [42], [43], 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 higher-order eukaryotic systems, adaptations of more sophisticated change detection schemes, and surveys of a broader range of sampling frequencies.

This publicly-available 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: \( \varepsilon_{P}^{NS} \) is used to denote the energy of the case in which RNAP is bound to a non-specific (NS) DNA binding site, \( \varepsilon_{P,1}^{S} \) the energy when RNAP is only bound to the \( S \) binding site, \( \varepsilon_{P,1}^{P} \) the energy when RNAP is specifically bound to the promoter-regulator complex, \( \varepsilon_{SS}^{S} \) the energy when the \( SS \) is bound to the \( NS \) binding site, \( \varepsilon_{SS}^{SS} \) the energy when the \( SS \) is bound to the \( S \) binding site, \( \varepsilon_{1}^{S} \) the energy when Regulator 1 is bound to the \( NS \) binding site, and \( \varepsilon_{1}^{SS} \) the energy when Regulator 1 is bound to the \( S \) binding site, and

\[
\Delta \varepsilon_{P,1}^{P} := e_{P}^{P,1} - e_{P}^{NS}, \quad \Delta \varepsilon_{P,1}^{SS} := e_{P}^{P,1} - e_{P}^{SS}, \quad \Delta \varepsilon_{1} := e_{1}^{S} - e_{1}^{NS}.
\]

Also define

\[
Z(P, R_1, R_{SS} - 1) := \frac{m! e^{-\pi e_{SS}^{S}} e^{-\pi e_{1}^{S}} e^{-\pi (R_{SS} - 1)} e^{-\pi e_{SS}^{SS}}}{P! R_1! (R_{SS} - 1)! (m - P - R_1 - R_{SS} + 1)!}
\]

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: \( Z(P, P - 1, R_{SS} - 1) e^{-\beta \varepsilon_{1}^{SS}} \), (iii) only RNAP bound: \( Z(P - 1, R_1, R_{SS} - 1) e^{-\beta \varepsilon_{P,1}^{SS}} \), and (iv) both Regulator 1 and RNAP bound: \( Z(P - 1, R_1, R_{SS} - 1) e^{-\beta \varepsilon_{P,1}^{P}} \). 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 [21], 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.

\[
p_{i}^{bound} = \frac{Z(P - 1, R_1, R_{SS} - 1) e^{-\beta \varepsilon_{P,1}^{P,10}} \times Z(P - 1, R_1, R_{SS} - 1) e^{-\beta \varepsilon_{P,1}^{P,11}} \times Z(P - 1, R_1, R_{SS} - 1) e^{-\beta \varepsilon_{P,1}^{SS,10}} \times Z(P - 1, R_1, R_{SS} - 1) e^{-\beta \varepsilon_{P,1}^{SS,11}}}{Z_{total}}
\]
The knot sequence is a set of points that divides a real interval into such as computational efficiency and numerical stability. Particularly, \( P \) where we have applied the approximation \( m!/P!R_1!(R_{SS} - 1)!(m - P - R_1 - R_{SS} + 1)! \approx m^m R_1^R_{SS} / P!R_1!(R_{SS} - 1)! \). 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.

where we have applied the approximation \( m!/P!R_1!(R_{SS} - 1)! \).

\( \Delta \varepsilon_{ij} \) is the binding energy of the \( j \)th complex to the promoter, \( \Delta \varepsilon_{P,i} \) is the energy of RNAP being bound to the promoter-regulator complex \( j \), and \( P \) is the concentration of RNAP. Setting \( a_{ij} = P e^{-\Delta \varepsilon_{P,i} e^{-\Delta \varepsilon_{ij}}} \) and \( b_{ij} = (1 + P e^{-\Delta \varepsilon_{P,i} e^{-\Delta \varepsilon_{ij}}} \), 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 [44].

A set of B-spline basis functions in variable \( t \) is determined by the degree of a piecewise polynomial, \( P \), and a knot sequence [45]. 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 \( t_0 \leq t_1 \leq \cdots \leq t_{D+P} \). Employing this set of knots and the De Boor recursion in [46], the \( d \)th B-spline basis of degree \( P \), written as \( \varphi_d^{(P)}(t) \), is derived recursively as follows:

\[
\varphi_d^{(0)}(t) = \begin{cases} 
1 & \text{if } t_{d-1} \leq t \leq t_d, \\
0 & \text{if otherwise}
\end{cases}
\]

\[
\varphi_d^{(p)}(t) = \frac{t - t_{d-1}}{t_{p+d} - t_{d-1}} \varphi_d^{(p-1)}(t) + \frac{t_{p+d} - t}{t_{p+d} - t_{d-1}} \varphi_{d+1}^{(p-1)}(t),
\]

for \( 1 \leq d \leq D + P - p \) where \( p = 0 \) in (21) and \( 1 \leq p \leq P \) in (22). The above recursion is visualized in Figure 8 (reconstructed from [45]).

\[
\begin{align*}
(t_0, t_1) & \quad \varphi_0^{(0)}(t) \\
(t_1, t_2) & \quad \varphi_0^{(1)}(t) \\
(t_2, t_3) & \quad \varphi_1^{(0)}(t) \\
(t_3, t_4) & \quad \varphi_1^{(1)}(t) \\
(t_4, t_5) & \quad \varphi_2^{(0)}(t) \\
(t_5, t_6) & \quad \varphi_2^{(1)}(t) \\
(t_6, t_7) & \quad \varphi_3^{(0)}(t)
\end{align*}
\]

Fig. 8. The De Boor recursion for \( P = 3 \) and \( D = 4 \).

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.

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 [36].

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

Definition 2. A function \( f(x, y) : B \rightarrow \mathbb{R} \) 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 [47]. The ADMM is an iterative augmented Lagrangian method that uses partial updates for dual variables and replaces joint minimization by simpler sub-problems. However, the mentioned procedure does not guarantee global optimality of the solution.

D. Proof of Theorem 1

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

\[
\nabla a_i \Gamma(\tilde{a}_i, \tilde{b}_i, \tilde{\lambda}_i) = 2 \sum_{l=1}^{L} \Omega_l(\tilde{a}_i, \tilde{b}_i, \tilde{\lambda}_i) p_l(t_i) + \gamma_2 \text{sign}(\tilde{a}_i) = 0
\]

\[
\nabla b_i \Gamma(\tilde{a}_i, \tilde{b}_i, \tilde{\lambda}_i) = -2 \sum_{l=1}^{L} \Omega_l(\tilde{a}_i, \tilde{b}_i, \tilde{\lambda}_i) u_i^T(t_i) \tilde{\lambda}_i p_l(t_i) + \gamma_1 \tilde{b}_i + \gamma_2 \text{sign}(\tilde{b}_i) = 0
\]

\[
\nabla \lambda_i \Gamma(\tilde{a}_i, \tilde{b}_i, \tilde{\lambda}_i) = -2 \sum_{l=1}^{L} \Omega_l(\tilde{a}_i, \tilde{b}_i, \tilde{\lambda}_i) p_l^T(t_i) \tilde{b}_i u_i(t_i) + \gamma_1 \tilde{\lambda}_i + \gamma_2 \text{sign}(\tilde{\lambda}_i) = 0
\]
with respect to constraints $0 \leq \bar{a}_i \leq \bar{b}_i$ and $\bar{\lambda}_i \geq 0$. These constraints admit that $\text{sign}(\cdot)$ can be replaced by vector $1$ in the above equations. It is obvious from (24)-(25) that $b_i^T \nabla_{b_i} \Omega(\bar{a}_i, \bar{b}_i, \bar{\lambda}_i) = \bar{\lambda}_i^T \nabla_{\bar{\lambda}_i} \Omega(\bar{a}_i, \bar{b}_i, \bar{\lambda}_i) = 0$, which results in

$$2 \sum_{l=1}^{L} \Omega(\bar{a}_l, \bar{b}_l, \bar{\lambda}_l)p_l^T(t_l)b_lu_l^T(t_l)\bar{\lambda}_l = \gamma_1 b_i^T \bar{b}_i + \gamma_2 \bar{\lambda}_i^T \mathbf{1}. \quad (26)$$

Consider the convex optimization

$$(P5) \min_{\{a_i, G_i, W_1, W_2\}} \sum_{l=1}^{L} \left( p_l^T(t_l)a_i - u_l^T(t_l)G_ip_l(t_l) \right)^2 + \gamma_1 \kappa(W_1, W_2) + \gamma_2 \|a\|_1$$

subject to $W := \begin{pmatrix} W_1 & G_i \\ G_i^T & W_2 \end{pmatrix} \succeq 0, \quad (27)$

where $\kappa(W_1, W_2) := \frac{1}{2} (\text{Tr}(W_1) + \text{Tr}(W_2))$. Minimizing (P5) with respect to $\{W_1, W_2\}$ leads to

$$\|G_i\|_* = \min_{\{W_1, W_2\}} \kappa(W_1, W_2) \quad \text{subject to} \quad W \succeq 0,$$

which is the alternative characterization of the nuclear norm [48]. Taking advantage of the nuclear norm, we can restrict matrix $G_i$ to be rank one as $\lambda_i b_i^T$. Also, $\kappa(\cdot, \cdot)$ is able to satisfy the required sparsity for $\{\lambda_i, b_i^T\}$. To investigate these claims, recall constraints (14) and set $G_i := \lambda_i b_i^T$, $W_1 := \lambda_i \lambda_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(\lambda_i)$, and $W_2 := b_i b_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(b_i)$ where $\text{diag}(\lambda_i)$ is the diagonal matrix with $(j,j)$ entry equal to $\lambda_j$. Then, the triple $(G_i, W_1, W_2)$ is feasible for (P5) due to

$$\begin{pmatrix} W_1 & G_i \\ G_i^T & W_2 \end{pmatrix} = \begin{pmatrix} \lambda_i \lambda_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(\lambda_i) & \lambda_i b_i^T \\ b_i \lambda_i^T & b_i b_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(b_i) \end{pmatrix} = \begin{pmatrix} \lambda_i & b_i^T \\ b_i & \lambda_i^T \end{pmatrix} + \frac{1}{\gamma_1} \begin{pmatrix} \gamma_3 \text{diag}(\lambda_i) & 0 \\ 0 & \gamma_2 \text{diag}(b_i) \end{pmatrix} \succeq 0. \quad (28)$$

In addition, we have

$$\gamma_1 \kappa(W_1, W_2) = \gamma_1 \left( \|\lambda\|_2^2 + \|b\|_2^2 \right) + \gamma_2 \|\lambda\|_1 + \gamma_3 \|b\|_1,$$

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

$$\mathcal{L}(G_i, a_i, W_1, W_2, M) = \sum_{l=1}^{L} \left( p_l^T(t_l)a_i - u_l^T(t_l)G_ip_l(t_l) \right)^2 \cdot \gamma_1 \kappa(W_1, W_2) - \langle M, W \rangle + \gamma_2 \|a\|_1,$$

and $M$ indicates the dual variable associated with the constraint $W \succeq 0$. In accordance with the block structure of $W$ 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

$$\nabla_{a_i} \mathcal{L}(G_i, a_i, W_1, W_2, M) =$$

$$2 \sum_{l=1}^{L} \left( p_l^T(t_l)a_i - u_l^T(t_l)G_ip_l(t_l) \right)^2 p_l(t_l) + \gamma_2 \text{sign}(a_i) \quad (29)$$

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

(iii) primal feasibility $W \succeq 0$;

(iv) dual feasibility $M \succeq 0$.

Consider the stationary points of (P3), and choose the candidate primal variables $a_i := a_i, G_i := \lambda_i b_i^T$, $W_1 := \lambda_i \lambda_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(\lambda_i)$, $W_2 := b_i b_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(b_i)$; and the dual variables $M_1 := \frac{\gamma_2}{\gamma_1} I$, $M_2 := \frac{\gamma_1}{\gamma_2} I$, $M_3 := - \sum_{l=1}^{L} \Omega_l(\bar{a}_l, \bar{b}_l, \bar{\lambda}_l)u_l(t_l)p_l^T(t_l)$, and $M_4 := M_{21}$. Then, condition (i) holds because the sub-gradients (29)-(32) are zero when substituting the introduced primal and dual variables. The requirement (ii) is also true since

$$\langle M, W \rangle = \langle M_1, W_1 \rangle + \langle M_2, W_2 \rangle + 2 \langle M_3, G_i \rangle \quad = \gamma_1 \frac{2}{\gamma_1} \text{Tr} \left( \lambda_i \lambda_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(\lambda_i) \right) + \gamma_2 \frac{2}{\gamma_1} \text{Tr} \left( b_i b_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(b_i) \right)$$

$$- 2 \text{Tr} \left( \sum_{l=1}^{L} \Omega_l(\bar{a}_l, \bar{b}_l, \bar{\lambda}_l)u_l(t_l)p_l^T(t_l) \right)$$

$$= \gamma_1 \frac{2}{\gamma_1} \text{Tr} \left( \lambda_i \lambda_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(\lambda_i) \right) + \gamma_2 \frac{2}{\gamma_1} \text{Tr} \left( b_i b_i^T + \frac{\gamma_2}{\gamma_1} \text{diag}(b_i) \right)$$

$$- \frac{1}{\gamma_1} \text{Tr} \left( \gamma_3 \text{diag}(\lambda_i) \right) - \frac{1}{\gamma_2} \text{Tr} \left( \gamma_3 \text{diag}(\lambda_i) \right) = 0,$$

where the last equality follows from (26). Moreover, (iii) is confirmed similar to (28). In order to meet the last criterion (iv), according to a Schar complement argument [34], it is sufficient to invoke $\|M_2\| \leq \gamma_1/2$. 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.

REFERENCES


Supplementary Figure 1. ROC and P-R curves for Dream 4, Challenge 2 Network 1 (top) and Network 2 (bottom).