Identification of Stable Genetic Networks using Convex Programming

Michael M. Zavlanos, A. Agung Julius, Stephen P. Boyd and George J. Pappas

Abstract— Gene regulatory networks capture interactions between genes and other cell substances, resulting in various models for the fundamental biological process of transcription and translation. The expression levels of the genes are typically measured in mRNA concentrations in micro-array experiments. In a so called genetic perturbation experiment, small perturbations are applied to equilibrium states and the resulting changes in expression activity are measured. This paper develops a novel algorithm that identifies a sparse stable genetic network that explains noisy genetic perturbation experiments obtained at equilibrium. Our identification algorithm can also incorporate a variety of possible prior knowledge of the network structure, which can be either qualitative, specifying positive, negative or no interactions between genes, or quantitative, specifying a range of interaction strength. Our method is based on a convex programming relaxation for handling the sparsity constraint, and therefore is applicable to the identification of genome-scale genetic networks.

I. Introduction

Recent advances in the field of biotechnology have targeted increasing interdisciplinary research into modeling and simulation of biological networks at the molecular level. In particular, the use of RNA micro-arrays that enables experimental gene expression measurements for large scale biological networks, has provided researchers with valuable data that can be used to identify gene interactions in large genetic networks. Besides promoting biological knowledge, identification of such networks is also important in drug discovery, where a systems-wide understanding of regulatory networks is crucial for identifying the targeted pathways.

Due to the significance of its potential applications, genetic network identification has recently received considerable attention. Depending on whether identification aims at relating the expression of a gene to the sequence motifs found in its promoter or to the expression of other genes in the cell, approaches can be characterized as *gene-to-sequence* or *gene-to-gene*, respectively [1], [2]. The ensemble of both classes consist the so called *genetic network identification* problem and solution techniques can either ignore or explicitly consider the underlying gene dynamics.

Members of the former class are clustering algorithms [3], [4] that group genes with similar expressions, due to the high probability that they are functionally, but not necessarily directly, related to each other. Alternatively, grouping of co-expressed genes may be achieved using information-theoretic methods [5]. Both approaches, however, are restricted to identifying undirected networks and hence, lack causality. Causality may be recovered using Bayesian networks [6], which can handle directed graphs. Nonetheless, Bayesian networks typically do not accommodate cycles and hence, can not handle feedback motifs that are common in genetic regulatory networks. Both causality and feedback motifs are no longer an issue when the network is modeled as a set of differential equations [7]–[10]. Identification is then typically optimization based, while approaches depend on whether the data is obtained from dynamic time-series or steady-state measurements.

The approach proposed in this paper falls under the latter class of networks modeled as differential equations and aims at obtaining a minimal model that explains given genetic perturbation data at steady-state. The minimality specification is due to the observation that biological networks exhibit loose connectivity [11], [12] and in the present framework, it was first addressed in [7] in the form of a priori combinations of constraints on the connectivity of the network. To avoid the combinatorially hard nature of the problem, in this paper, we employ a weighted ℓ_1 relaxation [13]–[16], which leads to a much more scalable linear program. Additional linear constraints are introduced so that our model best fits the given genetic perturbation data as well as satisfies a priori knowledge on the network structure. We show that the proposed linear program performs well for sufficiently large data sets with low noise, while smaller and noisy data sets hinder its performance, partly due to identification of unstable networks, which also contradict the steady-state assumption on the data.

The identification performance can, however, be greatly improved by imposing a stability condition on the identified network. This can be done using linear constraints, via Geršgorin's Theorem, or semidefinite ones, via a Lyapunov inequality ensuring stability. To the best of our knowledge, this is a first attempt to formally study the effect of stability of the identified networks on the identification performance.

The rest of this paper is organized as follows. In Section II we describe the genetic network identification problem, while in Section III we develop the proposed ℓ_1 relaxation and discuss the aforementioned stability issues that could hinder its identification performance. In Section IV we extend our algorithm to account for stability of the identified solutions and finally, in Section V, we illustrate the efficiency of our approach by testing it on artificial noisy data sets.

II. Genetic Network Identification

Genetic regulatory networks consisting of n genes can be modeled as n-dimensional dynamical systems [7]. In general, such models assume the form

$$\dot{\hat{x}} = f(\hat{x}, u), \tag{1}$$

where $\hat{x}(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^p$. Here $\hat{x}_i(t) \in \mathbb{R}$ denotes the transcription activity (typically measured as mRNA concentration) of gene i in the network, and u_i is the so called transcription perturbation.[1] Nonlinear genetic networks as in (1) can have multiple stable equilibria, each one typically corresponding to a phenotypical state of the system. Then, the dynamics in a neighborhood of any given equilibrium x_{eq} can be approximated by the set of linear differential equations

$$\dot{x} = Ax + Bu, \tag{2}$$

where $x \triangleq \hat{x} - x_{eq}$ [8]. The matrix $A \in \mathbb{R}^{n \times n}$ encodes pairwise interactions between the individual genes in the network at the given equilibrium or phenotypical state, while the matrix $B \in \mathbb{R}^{n \times p}$ indicates which genes are affected by the transcriptional perturbations. Assuming the equilibrium $x = 0$ is stable and the perturbation u is sufficiently small and constant, the system (2) will restabilize at a new equilibrium x, at which

$$Ax + Bu = 0. \tag{3}$$

Let m be the number of available transcription perturbations[2] and define the matrices $U = [u_1 \cdots u_m] \in \mathbb{R}^{p \times m}$ and $X = [x_1 \cdots x_m] \in \mathbb{R}^{n \times m}$ containing the transcription perturbations of all m experiments and their associated steady-state mRNA concentrations, respectively. Then, collecting all m experiments at steady-state, system (3) can be written as

$$AX + BU = 0. \tag{4}$$

The matrices X, U can be measured (possibly with noise) and are assumed to be known; the matrix B is also typically known. Then, the network identification problem can be stated as follows.

Problem 1 (Genetic Network Identification): Given steady-state transcription perturbation and mRNA concentration data X and U, determine the sparsest stable matrix A that *best* satisfies (4), while incorporating any a priori biological knowledge regarding the presence, absence, or nature of specific gene interactions.

The requirement that A is sparse is due to biological networks being sparse in nature [11], [12], while the stability condition is necessary for the steady-state to be observed. Finally, accordance with a priori biological knowledge is both desired and naturally expected to result in improved identification performance.

[1] For large scale networks, we may assume that not all genes are affected by a given perturbation, resulting in $p \leq n$.

[2] Typically, each transcription perturbation corresponds to a specific experiment.

III. Linear Programming Formulation

Given any genetic network described by (2), the problem of identifying the sparsest matrix A that approximately satisfies constraints (4), can be formulated as the following optimization problem

$$\begin{aligned} \text{minimize} \quad & \text{card}(A) \\ \text{subject to} \quad & \|AX + BU\|_1 \leq \epsilon, \end{aligned} \tag{5}$$

where $\text{card}(A)$ denotes the number of zero entries in matrix A, and $\|A\|_1 = \sum_{i,j=1}^n |a_{ij}|$ denotes the (elementwise) ℓ_1 norm of a matrix A. Variable in problem (5) consists the matrix A, while the problem data are X, B, U and ϵ. The positive parameter ϵ is used to control the trade-off between sparsity, i.e., $\text{card}(A)$, and best fit, i.e., $\|AX + BU\|_1$. Note that any other norm could be used in the constraints here; we use the ℓ_1 norm since it handles outliers well.

When a priori knowledge about the network is also available, it is typically in the form of a partial sign pattern $S = (s_{ij}) \in \{0, +, -, ?\}^{n \times n}$, which encodes known positive interactions $(+)$, negative interactions $(-)$, no interactions (0), or no a priori knowledge regarding interactions $(?)$ between any two genes in the network. Such knowledge can be included in (5) by means of the set of linear constraints

$$A \in S \quad \Rightarrow \quad \begin{cases} a_{ij} \geq 0, & \text{if } s_{ij} = + \\ a_{ij} \leq 0, & \text{if } s_{ij} = - \\ a_{ij} = 0, & \text{if } s_{ij} = 0 \\ a_{ij} \in \mathbb{R}, & \text{if } s_{ij} = ? \end{cases} \tag{6}$$

resulting in the problem

$$\begin{aligned} \text{minimize} \quad & \text{card}(A) \\ \text{subject to} \quad & \|AX + BU\|_1 \leq \epsilon, \quad A \in S. \end{aligned} \tag{7}$$

From a computational point of view, formulation (7) poses a significant challenge. Although both constraints are convex in the matrix A [17], the cost function $\text{card}(A)$ is not convex. Solving this problem globally can be done, for instance by branch-and-bound methods or direct solution with all possible 2^{n^2} sparsity patterns for A. Nevertheless, these methods are typically very slow, and cannot scale to networks with more than a handful of genes.

To obtain a method that can scale to large networks, we propose a convex relaxation of the cardinality cost function. In particular, we replace the $\text{card}(A)$ objective with the weighted ℓ_1-norm $\sum_{i,j=1}^n w_{ij}|a_{ij}|$, resulting in the following *convex* program

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j=1}^n w_{ij}|a_{ij}| \\ \text{subject to} \quad & \|AX + BU\|_1 \leq \epsilon, \quad A \in S, \end{aligned} \tag{8}$$

where the weights w_{ij} are chosen such that

$$w_{ij} = \frac{\delta}{\delta + |a_{ij}|}, \quad \text{for all } i, j = 1, \dots, n \tag{9}$$

for sufficiently small $\delta > 0$ [14]. The main idea behind the proposed heuristic is to uniformly initialize all weights by $w_{ij} = 1$ (this corresponds to the standard ℓ_1 relaxation of the cost function) and repeatedly solve problem (8), each time updating the weights using (9) (Algorithm 1). Then, large

Algorithm 1 Network Identification (Ignoring Stability)

Require: Sign pattern S and possibly noisy experimental data $X \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{p \times m}$,

1: Set $\epsilon \triangleq \mu \min\{\|AX + BU\|_1 \mid A \in S\}$, with $\mu > 1$,
2: Initialize weights $w_{ij} = 1$ for all $i, j = 1, \ldots, n$,
3: **for** $it = 1$ to J **do**
4:     Solve the linear program (8),
5:     Update the weights w_{ij} using (9),
6: **end for**

Correct Positives Identified[†]	1
Correct Negatives Identified[†]	4
Correct Zeros Identified[†]	4
False Positives Identified	3
False Negatives Identified	4
False Zeros Identified	1

† a priori sign knowledge not included.

weights are always assigned to small matrix entries $|a_{ij}|$ and small weights to large entries $|a_{ij}|$, which can eliminate any weak genetic interactions in the final identified matrix A.

In practice, Algorithm 1 requires no more than $J = 10$ iterations, regardless of the problem size. Furthermore, recent theoretical results [18] show that, in some cases (not including the present application), minimizing the weighted ℓ_1 norm of a matrix A, in fact does minimizes $\text{card}(A)$ with high probability. Despite, however, the computationally appealing properties of Algorithm 1, its identification performance can be significantly decreased in the presence of noisy data or in the absence of a sufficient number of experiments m. We illustrate this observation in the following example and argue that it might be due to instability of the identified network.

Example 3.1 (Identifying an unstable matrix): Consider the following 5×5 matrix A to be identified

$$A = \begin{bmatrix} -1.2294 & -1.5474 & 0 & 0 & 0 \\ 0.4153 & -0.8308 & 0 & 0 & 0 \\ 0.3614 & 0 & -0.5833 & 0 & 1.0243 \\ 0 & 0 & 0 & -0.9589 & -0.3565 \\ 0.0420 & 0 & 0 & 0 & -1.3703 \end{bmatrix}$$

and assume the desired a priori sign pattern is given by

$$S = \begin{bmatrix} ? & -1 & ? & ? & ? \\ ? & ? & 0 & ? & 0 \\ ? & ? & ? & 0 & 1 \\ ? & ? & 0 & ? & ? \\ 1 & ? & ? & 0 & ? \end{bmatrix}.$$

Let $m = 4$ be the number of available experiments and assume a significantly high noise level of 80%. Using these quantities we can construct noisy data by $X = -A^{-1}BU + 0.8N$, where $B = I_5$, $U = 0.1I_{5 \times 4}$ and $N \in \mathbb{R}^{5 \times 4}$ is a zero mean and unit variance normally distributed random variable.[3] Applying $J = 10$ iterations of Algorithm 1, the identified matrix A_u becomes

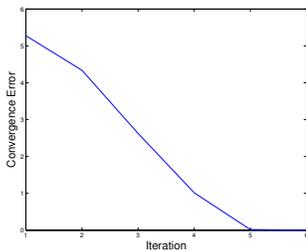[3]We denote by $I_{n \times m}$ the $n \times m$ identity matrix.

Fig. 1. Plot of the convergence error $\|A_{k+1} - A_k\|_1$ as a function of the iteration k of Algorithm 1 for Example 3.1.

$$A_u = \begin{bmatrix} -0.5142 & -1.2042 & 0 & 0 & -0.4896 \\ 0.3685 & -0.8354 & 0 & 0 & 0 \\ -0.2332 & -0.0689 & -0.2916 & 0 & 8.7007 \\ 0.0151 & 0.0018 & 0 & 2.5766 & 0 \\ 0.2439 & -0.0306 & 0 & 0 & -8.6554 \end{bmatrix}.$$

Fig. 1 shows the convergence rate of Algorithm 1. Clearly, the desired sign pattern S is satisfied and A_u is a sparse enough matrix. Note, however, that A_u is not a stable matrix even though A is stable. Table I summarizes the performance of Algorithm 1 in identifying matrix A. Note that the number of false positives, negatives and zeros in the identified matrix A_u is significantly large, leading to the conclusion that stability of A_u might be an important aspect of the problem.

IV. INCORPORATING STABILITY

In Section III we developed an iterative procedure, based on the solution of linear programs, able to identify a sparse matrix that best fits possibly noisy network data, while satisfying a priori knowledge about the network. Despite these appealing properties of the proposed approach, it was also shown in Example 3.1 that it is possible that the identified matrix is unstable, violating the specifications of Problem 1 and hindering the algorithm's performance. In this section, we propose two different ways of incorporating stability in Algorithm 1, both preserving its convex nature and hence, having the associated scalability and global optimality properties. Furthermore, we show that these modified approaches significantly increase the performance of our identification algorithm.

A. Linear Approximation

Incorporating stability of the identified matrix A as a linear constraint in Algorithm 1 relies on the following theorem by Geršgorin.

Theorem 4.1 ([19]): Let $A = (a_{ij}) \in \mathbb{C}^{n \times n}$ and for all $i = 1, \ldots, n$ define the *deleted absolute row sums* of A by $R_i(A) \triangleq \sum_{j \neq i} |a_{ij}|$. Then, all eigenvalues of A are located in the union of n discs $G(A) \triangleq \cup_{i=1}^{n} \{z \in \mathbb{C} \mid |z - a_{ii}| \leq R_i(A)\}$. Furthermore, if a union of k of these n discs forms a connected region that is disjoint from all the remaining $n - k$ discs, then there are exactly k eigenvalues of A in this region.

The region $G(A)$ is often called the *Geršgorin region* (for the rows) of A, the individual discs in $G(A)$ are called the *Geršgorin discs*, while the boundaries of these discs are called the *Geršgorin circles*. Since A and A^T have the same

Algorithm 2 Network Identification (Geršgorin Stability)

Require: Sign pattern S and possibly noisy experimental data $X \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{p \times m}$,
 1: Set $\epsilon \triangleq \mu \min\{\|AX + BU\|_1 \mid A \in S\}$, with $\mu > 1$,
 2: Initialize weights $w_{ij} = 1$ for all $i, j = 1, \ldots, n$,
 3: **for** $it = 1$ to J **do**
 4:     Solve the linear program (12),
 5:     Update the weights w_{ij} using (9),
 6:     If $it = 1$, update the weights v_i using (13),
 7: **end for**

---

eigenvalues, one can also obtain a similar Geršgorin disc theorem for the columns of A. Clearly, if

$$a_{ii} \leq -\sum_{j \neq i} |a_{ij}|, \quad \text{for all } i = 1, \ldots, n \qquad (10)$$

then all discs $\{z \in \mathbb{C} \mid |z - a_{ii}| \leq R_i(A)\}$ are in the left half plane $\mathbb{C}_-$ and Theorem 4.1 ensures that all eigenvalues of A are also in $\mathbb{C}_-$, which implies that A is stable. What is appealing about constraints (10) is that they are linear in the entries of A and, hence, can be directly incorporated in the linear program (8) in Algorithm 1, rendering a stable matrix. However, constraints (10) also impose strict structural constraints on the entries of A. In particular, they restrict all diagonal entries of A to be non-positive and matrix A to be *diagonally dominant*, namely $|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$ for all $i = 1, \ldots, n$. This constraint can be relaxed by applying a similarity transformation on A. In particular, since $V^{-1}AV$ and A share the same eigenvalues for any invertible matrix V, we can apply Geršgorin's theorem to $V^{-1}AV$ and for a smart choice of V we can obtain sharper bounds on the eigenvalues. A particularly convenient choice is $V \triangleq \mathrm{diag}(v_1, \ldots, v_n)$, with $v_i > 0$ for all $i = 1, \ldots, n$. Then, $V^{-1}AV = (v_j a_{ij}/v_i)$ and Geršgorin's theorem states that all eigenvalues of A lie in the region

$$G(V^{-1}AV) \triangleq \cup_{i=1}^n \left\{ z \in \mathbb{C} \mid |z - a_{ii}| \leq \frac{1}{v_i} \sum_{j \neq i} v_j |a_{ij}| \right\}.$$

Clearly, if we require that

$$a_{ii} \leq -\frac{1}{v_i} \sum_{j \neq i} v_j |a_{ij}|, \quad i = 1, \ldots, n, \qquad (11)$$

then $G(V^{-1}AV) \subset \mathbb{C}_-$, which implies that matrix A is stable, but not necessarily diagonally dominant any more. Constraints (11) are still convex in the entries of A and hence, can be directly incorporated in (8) resulting in the linear program

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j=1}^n w_{ij} |a_{ij}| \\
\text{subject to} \quad & \|AX + BU\|_1 \leq \epsilon, \quad A \in S \\
& a_{ii} \leq -\frac{1}{v_i} \sum_{j \neq i} v_j |a_{ij}|, \quad i = 1, \ldots, n.
\end{aligned}
\qquad (12)
$$

The identification procedure is then described in Algorithm 2. Intuitively, the weights v_i, should penalize Geršgorin discs far in the left half plane and assign the remaining slack to discs close to the imaginary axis, breaking in this way the

diagonal dominance in the associated row. In particular, for any $\beta > 0$ we choose the weights v_i by

$$
v_i \triangleq \begin{cases} \beta + \frac{|a_{ii}| - R_i(A)}{1 + |a_{ii}| - R_i(A)}, & \text{if } |a_{ii}| - R_i(A) > 0 \\ \beta, & \text{if } |a_{ii}| - R_i(A) \leq 0 \end{cases}, \qquad (13)
$$

where $R_i(A)$ denotes the deleted absolute sum for row i, as in Theorem 4.1, and the quantity $|a_{ii}| - R_i(A) > 0$ indicates how far in the left half plane the associated Geršgorin disc is located. Note that the weights v_i are updated only once (line 6 in Algorithm 2) so that Algorithm 2 possesses similar convergence properties with Algorithm 1. The following example illustrates the performance of Algorithm 2 for the data in Example 3.1.

Example 4.2 (Stable Identification using Algorithm 2): Applying Algorithm 2 to the data given in Example 3.1 the identified matrix A_s becomes

$$
A_s = \begin{bmatrix} -0.5356 & -1.1981 & 0 & 0 & 0 \\ 0.2499 & -1.1036 & 0 & 0 & 0 \\ 0.0081 & -0.0922 & -0.2938 & 0 & 0.1843 \\ -0.0016 & 0 & 0 & -0.3190 & 0 \\ 0.6089 & -0.0741 & 0 & 0 & -21.4519 \end{bmatrix}.
$$

Clearly, the desired sign pattern S is satisfied, while A_s is also sparse, stable and not diagonally dominant. Table II summarizes the performance of Algorithm 2. Note that the number of false positives, negatives and zeros in the identified matrix A_s is significantly reduced compared to the results in Table I, which implies that stability is indeed an important identification specification.

B. Semidefinite Approximation

Let A be the matrix identified by Algorithm 1 which, as shown in Example 3.1, can possibly be unstable. The goal in this section is to characterize "small" perturbations to A that render it stable, while satisfying the desired sign pattern and maintaining its sparsity structure. For this, let $D \in \mathbb{R}^{n \times n}$ be the sought perturbation matrix and define the matrix $A' \triangleq A + D$. A necessary and sufficient condition for stability of A' is the existence of a symmetric positive definite Lyapunov matrix P such that

$$(A + D)^T P + P(A + D) \prec 0. \qquad (14)$$

Letting $L \triangleq PD$, equation (14) becomes

$$A^T P + L^T + PA + L \prec 0, \qquad (15)$$

which is a linear matrix inequality in both P and L and can be efficiently solved using semidefinite programming [17].

Algorithm 3 Network Identification (Lyapunov Stability)

Require: Sign pattern S and possibly noisy experimental data $X \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{p \times m}$,
1: Apply Algorithm 1 for matrix A,
2: **if** matrix A is unstable **then**
3:     Solve (16) for stable matrix $A' = A + P^{-1}L$,
4:     Solve (17) for robust Lyapunov matrix $P^\star$,
5:     Set $A := A'$,
6:     Set $\epsilon \triangleq \mu \min\{\|AX + BU\|_1 \mid A \in S\}$, with $\mu > 1$,
7:     Initialize weights $w_{ij} = 1$ for all $i, j = 1, \dots, n$,
8:     **for** $it = 1$ to J **do**
9:         Solve the semidefinite program (20),
10:         Update the weights w_{ij} using (9),
11:     **end for**
12: **end if**

In particular, solving the following semidefinite program

$$
\begin{aligned}
\text{minimize} \quad & \|LX\|_2 \\
\text{subject to} \quad & A^T P + L^T + PA + L \preceq 0, \quad P \succeq I,
\end{aligned} \tag{16}
$$

gives $D = P^{-1}L$ and the desired stable matrix A' becomes $A' = A + P^{-1}L$. Since $\|(A' - A)X\|_2 = \|P^{-1}LX\|_2 \leq \frac{\|LX\|_2}{\|P\|_2}$ and $\|P\|_2 > 1$,[4] minimizing the objective $\|LX\|_2$ also minimizes the error between $AX + BU$ and $A'X + BU$. Clearly, matrix A' may no longer satisfy the desired sign pattern or sparsity specifications hence, we need to further perturb it while maintaining its stability. For this, we need to define the notion of a robust Lyapunov function.

Definition 4.3 (δ-robust Lyapunov function): We call P a δ-robust Lyapunov function of the stable system A if for any perturbation $\|\Delta\|_2 < \delta$, the system $A + \Delta$ is also stable with corresponding Lyapunov function P.

Obtaining a δ-robust Lyapunov function for the stable matrix A' relies on the solution of the following convex optimization problem

$$
\begin{aligned}
\text{minimize} \quad & \|P\|_2^2 - \mu, \\
\text{subject to} \quad & A'^T P + PA' \preceq -\mu I, \quad P \succ 0, \quad \mu > 0.
\end{aligned} \tag{17}
$$

The idea is that the matrix P obtained by (17) should robustly satisfy the Lyapunov equation by a positive constant $\mu > 0$. Let $P^\star$ and $\mu^\star$ be the solution to problem (17). Then, we have the following result.

Proposition 4.4 (Geometric Interpretation of Robustness): The Lyapunov function $P^\star$ of the system A' is δ-robust with $\delta = \frac{\mu^\star}{2\|P^\star\|_2}$.

Proof: Our goal is to find $\delta > 0$ such that

$$
(A' + \Delta)^T P^\star + P^\star(A' + \Delta) \prec 0, \tag{18}
$$

for any perturbation $\|\Delta\|_2 < \delta$. Let $Q = \Delta^T P^\star + P^\star \Delta$. Then, equation (18) becomes

$$
Q - \mu^\star I \prec 0, \tag{19}
$$

---
[4]Observe that $P \succ I$ implies that $1 < \lambda_{min}(P)$ and so $1 < \|P\|_2$ since, $\lambda_{min}(P) < \max_i\{|\lambda_i(P)|\} = \rho(P) \leq \|P\|_2$, where $\rho(P)$ denotes the spectral radius of P.

TABLE III

PERFORMANCE OF ALG. 3 IN EXAMPLE 4.5

Correct Positives Identified†	2
Correct Negatives Identified†	5
Correct Zeros Identified†	7
False Positives Identified	0
False Negatives Identified	2
False Zeros Identified	1

† a priori sign knowledge not included.

since $A'^T P^\star + P^\star A' \prec -\mu^\star I$, by the semidefinite program (17). A sufficient condition for equation (19) to be true is that $\lambda_{max}(Q) < \mu^\star$. But $\max_i |\lambda_i(Q)| = \rho(Q) \leq \|Q\|_2 \leq 2\|\Delta\|_2\|P^\star\|_2$ by the triangle inequality and clearly $\lambda_{max}(Q) \leq \rho(Q)$. Hence, a sufficient condition for equation (19) to hold is that $2\|\Delta\|_2\|P^\star\|_2 < \mu^\star$, which completes the proof. ∎

Proposition 4.4 characterizes the allowable perturbations to the matrix A' that do not violate its stability. This result enables us to safely search within a δ-neighborhood of the matrix A' for a sparser matrix that also satisfies the desired sign pattern. In particular, we can extend (8) by

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j=1}^{n} w_{ij}|a_{ij}| \\
\text{subject to} \quad & \|AX + BU\|_1 \leq \epsilon \\
& A^T P^\star + P^\star A \preceq 0, \quad A \in S
\end{aligned} \tag{20}
$$

and iterate until convergence, as in Algorithm 1. This procedure is described in Algorithm 3.

The following example illustrates the performance of Algorithm 3 for the data in Example 3.1.

Example 4.5 (Stable Identification using Algorithm 3): Applying Algorithm 3 to the data given in Example 3.1 the identified matrix A_s becomes

$$
A_s = \begin{bmatrix} -0.5277 & -1.2015 & 0 & 0 & 0 \\ 0.3685 & -0.8347 & 0 & 0 & 0 \\ 0.0025 & -0.0974 & -0.2915 & 0 & 0.3982 \\ 0 & 0 & 0 & -0.0064 & 0 \\ 0.0215 & -0.0017 & 0 & 0 & -0.7259 \end{bmatrix}.
$$

Clearly, the desired sign pattern S is satisfied, while A_s is also sparse, stable and not diagonally dominant. Table III summarizes the performance of Algorithm 3. Note that the number of false positives, negatives and zeros in the identified matrix A_s is significantly reduced compared to the results in Table I, which implies that stability is indeed an important identification specification.

V. RESULTS AND DISCUSSION

Example 3.1 indicates that partial data, i.e. $m < n$, and high noise levels in the system might hinder the identification performance of Algorithm 1, which, however, can be recovered if stability of the identified network is incorporated in the identification procedure, as shown in Examples 4.2 and 4.5. The goal in this section is to explore the identification capabilities of Algorithms 1, 2 and 3 for various problem sizes n, data set sizes m and noise levels and characterize the regions where each one performs best.

In particular, we test Algorithms 1, 2 and 3 for problems of size $n = 5, 10, 20$, data sets of size $m = n$ (Full Data) and
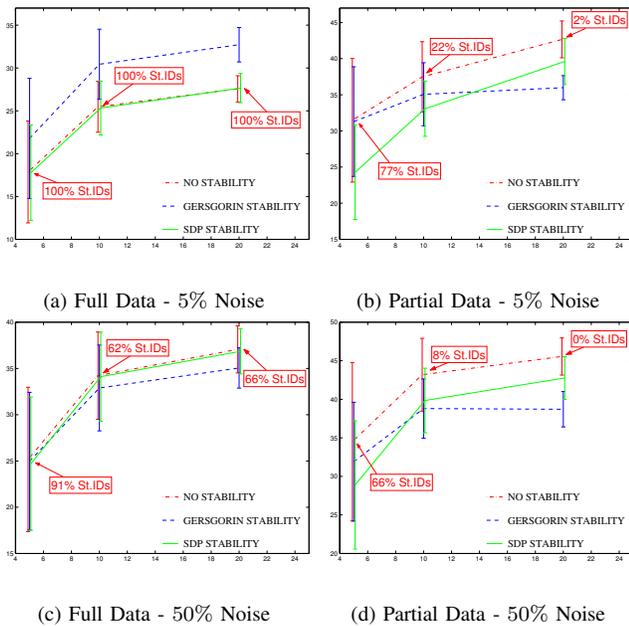
(a) Full Data - 5% Noise   (b) Partial Data - 5% Noise

(c) Full Data - 50% Noise   (d) Partial Data - 50% Noise

Fig. 2. Plots of the average total False Identifications (F.IDs) as a function of the Problem Size $n = 5, 10, 20$ for data sets of size $m = n$ (Full Data) and $m = \lceil \frac{n}{2} \rceil$ (Partial Data) and noise levels $\nu = 5\%$ and $\nu = 50\%$. Figs. 2(a)-2(b) and 2(c)-2(d) compare Algorithms 1, 2 and 3 for different data set sizes m and the same noise level ν, while Figs. 2(a)-2(c) and 2(b)-2(d) compare Algorithms 1, 2 and 3 for different noise levels ν and the same data set size m. Note also Stable Identifications (St.IDs) returned by Alg. 1.

$m = \lceil \frac{n}{2} \rceil$ (Partial Data) and noise levels $\nu = 5\%$ and $\nu = 50\%$. For every tuple (n, m, ν), a set of 50 *stable* matrices A to be identified is randomly generated from the normal distribution, such that every gene is connected to 60% of the genes in the network. The associated data sets are obtained by $X = -A^{-1}BU + \nu N$, where $B = I_n$, $U = 0.1I_{n \times m}$ and $N \in \mathbb{R}^{n \times m}$ is a zero mean and unit variance normally distributed random matrix. The a priori sign pattern used involves 30% of all gene interactions modeled in each matrix A. All algorithms are implemented in MATLAB using the `cvx` toolbox for convex optimization problems [20] and run on an Intel Xeon 2.8GHz processor with 4GB RAM. For problems of size $n = 20$, each iteration of Algorithms 1, 2 and 3 takes approximately 4, 6 and 10 seconds, respectively, while no more than 10 iterations are in general required.

Figs. 2 show how False Identifications (F.IDs) vary as a function of the problem size n for different pairs (m, ν) of data set sizes and noise levels. Note that Algorithm 1 performs best for full data and low noise levels, where it also returns 100% stable identifications (St.IDs), Algorithm 2 for partial data or high noise levels and large size problems and Algorithm 3 for partial data or high noise levels and small size problems. Note also that for large problems, best identifications are achieved by means of linear programming (Alg. 1 or Alg. 2) rather than semidefinite programming (Alg. 3), significantly reducing computational complexity.

VI. CONCLUSIONS

In this paper, we considered the problem of identifying a minimal model that best explains genetic perturbation data obtained at the network's equilibrium state. We relaxed the combinatorially hard cardinality optimization specification by employing its weighted ℓ_1 approximation and extended our formulation to account for a priori knowledge on the network structure, as well as stability of the derived solutions. We tested our algorithms on different size data sets corrupted by various noise levels and showed that stability of the identified network is an important aspect of the problem that can significantly increase the performance of identification algorithms, especially when partial and noisy data sets are considered. The strength of our approach lies in its convex nature that can handle large scale identification problems.

REFERENCES

[1] M. Bansal, V. Belcastro, A. Ambesi-Impiombato and D. di Bernardo. *How to Infer Gene Networks from Expression Profiles*, Molecular Systems Biology, vol. 3, 2007. (10.1038/msb4100120).

[2] T. Gardner and J. Faith. *Reverse-Engineering Transcription Control Networks*, Physics of Life Reviews, vol. 2, pp. 65-88, 2005.

[3] M. Eisen, P. Spellman, P. Brown, D. Botstein. *Cluster Analysis and Display of Genome-Wide Expression Patterns*, Proc. National Academy of Science, vol. 95, pp. 14863-14868, 1998.

[4] R. Amato, A. Ciaramella, N. Deniskina, C. Del Mondo, D. di Bernardo, C. Donalek, G. Longo, G. Mangano, G. Miele, G. Raiconi, A. Staiano and R. Tagliaferri. *A Multi-Step Approach to Time Series Analysis and Gene Expression Clustering*, Bioinformatics, vol. 22, pp. 589-596, 2006.

[5] R. Steuer, J. Kurths, C.O. Daub, J. Weise and J. Selbig. *The Mutual Information: Detecting and Evaluating Dependencies between Variables*, Bioinformatics, vol. 182, pp. 231-240, 2002.

[6] D. Pe'er, I. Nachman, M. Linial and N. Friedman. *Using Bayesian Networks to Analize Expression Data*, Journal of Computational Biology, vol. 7, pp. 601-620, 2000.

[7] T. Gardner, D. di Bernardo, D. Lorenz and J. Collins. *Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling*, Science, vol. 301, pp. 102-105, 2003.

[8] E. Sontag, A. Kiyatkin and B. Kholodenko. *Inferring Dynamic Architecture of Cellular Networks using Time Series of Gene Expression, Protein and Metabolite Data*, Bioinformatics, vol. 20(12), pp. 1877-1886, 2004.

[9] F. Amato, C. Cosentino, W. Curatola and D. di Bernardo. *LMI-based Algorithm for the Reconstruction of Biological Networks*, Proc. American Control Conference, pp. 2720-2725, New York, NY, 2007.

[10] A. Papachristodoulou and B. Recht. *Determining Interconnections in Chemical Reaction Networks*, Proc. American Control Conference, pp. 4872-4877, New York, NY, 2007.

[11] M. Arnone and E. Davidson. *The Hardwiring of Development: Organization and Function of Genomic Regulatory Systems*, Development, vol. 124.

[12] D. Thieffry, A. Huerta, E. Pérez-Rueda and J. Collado-Vides. *From Specific Gene Regulation to Genomic Networks: A Global Analysis of Transcriptional Regulation in Escherichia Coli*, Bioessays, vol. 20.

[13] E. J. Candes, M. B. Wakin and S. Boyd. *Enhancing Sparsity by Reweighted ℓ_1 Minimization*, Manuscript, Oct. 2007. Available at http://www.stanford.edu/~boyd/papers.html.

[14] S. Boyd. *ℓ_1-norm Methods for Convex Cardinality Problems*, Lecture Notes for EE364b, Stanford University. Available at http://www.stanford.edu/class/ee364b/.

[15] A. Hassibi, J. How and S. Boyd. *Low-Authority Controller Design via Convex Optimization*, AIAA Journal of Guidance, Control, and Dynamics, vol. 22(6), pp. 862-872, 1999.

[16] J. A. Tropp. *Just Relax: Convex Programming Methods for Identifying Sparse Signals*, IEEE Trans. Inf. Theory, vol. 51(3), pp. 1030-1051, 2006.

[17] S. Boyd and L. Vandenberghe. *Convex Optimization*, Cambridge U. Press, 2004.

[18] E. Candes, J. Romberg and T. Tao. *Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information*, IEEE Trans. Inf. Theory, vol. 52(2), pp. 489-509, 2006.

[19] R. Horn and C. Johnson. *Matrix Analysis*, Cambridge U. Press, 1985.

[20] M. Grant, S. Boyd, and Y. Ye, `cvx` *MATLAB Software for Disciplined Convex Programming*, Available at http://www.stanford.edu/~boyd/cvx/.