Proceedings of the 18th World Congress
The International Federation of Automatic Control
Milano (Italy) August 28 - September 2, 2011

Genetic Regulatory Network Identification Using Monotone Functions Decomposition [⋆]

A. Agung Julius   Calin Belta

Abstract: We present a method for identification of gene regulatory network topology using a time series of gene expression data. The underlying assumption in our method is that the functions that describe regulatory relations must be continuous, nonnegative and monotonic. This assumption is very general, as it is satisfied by virtually all existing regulatory models. Our method is based on refuting all regulation hypotheses that cannot meet this assumption. This procedure takes the form of a Linear Programming (LP) feasibility problem. We also present two conditions where the regulation hypotheses are irrefutable.

1. INTRODUCTION

Gene-gene interactions form the core of regulatory functions in cellular activities. The expression levels of various genes modulate cellular functions such as metabolism, cell division, programmable cell death (apoptosis), and intercellular signaling. The availability of high throughput measurement techniques presents biologists with a large quantity of data, organized as genome wide snapshots of gene expression activities. Expression levels are typically measured as transcript concentrations with DNA microarray (c.f. Bonneau et al. [2006], Faith et al. [2007]). One of the biggest challenges in systems biology is *to identify the interaction topology between the genes based on the expression activities data*. In short, this problem is often referred to as identification or reverse engineering of genetic regulatory networks (GRNs).

With the availability of such gene expression measurements, the network can in principle be reconstructed by inverting the data (c.f. Sontag et al. [2004]). However, as the measurement is noisy, special care needs to be taken to avoid inundating the reconstructed network with spurious interconnections. This concern gives rise to sparse identification or parsimonious identification that aims at getting a network model with as few connections as possible without losing the fitness to the data (c.f. Yeung et al. [2002], Gardner et al. [2003], Bonneau et al. [2006]).

Within the systems and control community, identification of GRNs in general and sparse identification of GRNs in particular are quite active research areas. For example, de Jong *et al* developed a method for identification of GRNs using the structure of piecewise affine dynamical systems (c.f. Drulhe et al. [2008], Porreca et al. [2008]). Papachristodoulou *et al* developed a model for identification of sparse networks using Hill functions to describe the dynamics of gene-gene interaction (c.f. August and Papachristodoulou [2009]). Earlier work by the first author also aimed at identifying sparse networks based on genetic perturbation data, assuming that the dynamics can be described (locally) as a linear system (c.f. Zavlanos et al. [2008], Julius et al. [2009]).

The method presented in this paper is different from the above references, in the sense that although we still use the dynamical system formulation, we do not rely on the structure of the underlying regulatory dynamics (e.g. linear, polynomial, Hill functions, etc.). The only assumption that we make is that the interaction dynamics can be represented as nonnegative monotonic functions. The network structure is built by identifying the set of regulators for each gene. The regulators of a gene X are the genes that directly [1] regulate the expression activity of X. Another point of departure from our previous work (c.f. Zavlanos et al. [2008], Julius et al. [2009]) is that the current method is essentially used for model invalidation, rather than model identification. In parallel with the development of the results in this paper, a recent work by Porreca et al (c.f. Porreca et al. [2010]) is published. They proposed a two-staged process in identifying a continuous-time differential equation model for GRNs. In the first stage, network topologies that are inconsistent with the data are rejected. This first stage is very similar to our approach, in the sense that it separates the issues of network topology and the functional/parametric representation of the dynamics. This paper differs from Porreca et al. [2010] in that, (i) we propose a discrete-time model structure that is directly derived from the time series data, (ii) we prove that the conditions that we use to reject some network topologies are both necessary and sufficient (see Theorem 2), while in Porreca et al. [2010], no such result is derived, (iii) we present some theoretical analysis on irrefutable models and data.

The main result of this paper is as follows. For any proposed set of regulators, we derive a necessary and sufficient condition for the data to be compatible with the regulators. This condition is formulated as a Linear Programming (LP) feasibility problem, which is computationally tractable. By verifying the compatibility of the regulators set with the data, we can invalidate some of the model structures. Another nice feature of our method is that it

[⋆] A. Agung Julius is with the Dept. Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. Calin Belta is with the Dept. Mechanical Engineering and Div. Systems Engineering, Boston University, Boston, MA. Email: agung@ecse.rpi.edu, cbelta@bu.edu

[1] Note that by *directly* we mean without going through other genes in the network under study. Therefore, this is not necessarily a statement about the binding of transcription factors to certain promoters.

978-3-902661-93-7/11/$20.00 © 2011 IFAC

10.3182/20110828-6-IT-1002.02181

is highly parallelizable. Indeed, the calculation of the set of regulators for each gene is independent from another, and therefore can be executed in parallel, e.g. on different processors. We also analyze some irrefutability results for certain network topologies and time series data. This result can be used in, e.g. determining whether the data is rich enough to separate different model structures.

2. MATHEMATICAL MODELS FOR GENE-GENE INTERACTION

We assume that we are working with a GRN with G genes. Also, we assume that we have a sequence of expression activities of the G genes, which are given at $(N+1)$ time points. We denote the data as $x_{i,j}$, $1 \le i \le G$, $0 \le j \le N$, which stands for the expression activity of Gene i at time j. We also define the (time) differential expression activities $q_{i,j}$, $1 \le i \le G$, $0 \le j \le N-1$, as

$$q_{i,j} \triangleq x_{i,j+1} - x_{i,j}. \tag{1}$$

To capture the interaction between genes, we adopt the following discrete-time dynamical system model:

$$q_{i,j} = -\lambda_i x_{i,j} + \sum_{k \in G_i^R} f_{i,k}(x_{k,j}), \tag{2}$$

where $\lambda_i \ge 0$ is the decay parameter of Gene i, $G_i^R \subseteq \{1,\dots,G\}$ is the set of regulators of Gene i, and $f_{i,k}$ is the function that describes the regulatory role of Gene k on Gene i. All these parameters are unknown and need to be identified.

Note that Eqn. (2) can be seen as a (first order, Euler) discrete approximation of a continuous-time differential equation for x_i, where we use a constant sampling period $\Delta t = t_{j+1} - t_j$, $j = 1, 2, \dots,$ and we denote t_j by j for simplicity. It is important to note that we do not assume that the set of Eqn. (2) are only valid locally around an equilibrium point, as is the case in the majority of gene reconstruction methods based on "small" perturbation.

The set of regulators G_i^R can be decomposed into two disjoint sets

$$G_i^R = G_i^{R+} \cup G_i^{R-}, \tag{3}$$

where G_i^{R+} and G_i^{R-} are the set of activators and repressors of Gene i, respectively. The number of regulators of a gene is called the in-degree of the gene, which is obvious from the graph-theoretic interpretation of the interaction network.

We adopt the following assumption for the regulatory function $f_{i,k}(\cdot)$.

Assumption 1: The functions $f_{i,k}(\cdot)$ are continuous, nonnegative and monotone, for all $i \in \{1,\dots,G\}$ and $k \in G_i^R$.

When $f_{i,k}(\cdot)$ is monotonically increasing, the interpretation is that Gene k is an *activator* of Gene i. On the other hand, when $f_{i,k}(\cdot)$ is monotonically decreasing, Gene k is a *repressor* of Gene i.

Remark 1. The assumption that f is continuous, nonnegative and monotone is very general. Virtually all phenomenological regulation models that have been proposed to represent gene-gene interaction (see references in Section 1) are captured in this broad class of functions.

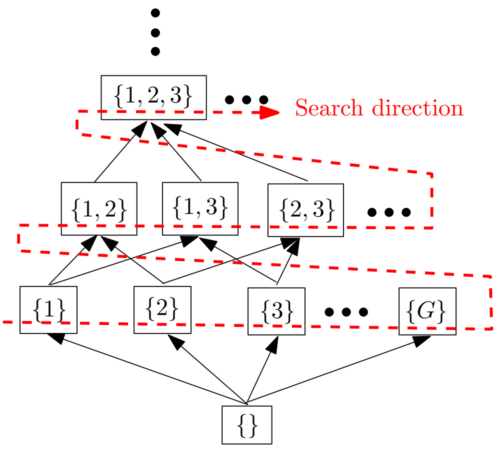

Fig. 1. The lattice structure of the power set of $\{1,\dots,G\}$. The arrows indicate set inclusion, and the direction of the breadth-first search is given by the dashed red line.

Hereafter, we shall refer to continuous, nonnegative and monotone functions as CNM functions, for brevity. Similarly, the increasing and decreasing types of CNM functions will be referred to as CNM+ and CNM- functions, respectively.

3. SPARSITY CONSIDERATION

We characterize the sparsity of a genetic regulatory network by the in-degree of the nodes (genes) in the network. Biologically, this means that we characterize the sparsity of the network by the number of regulators for each gene. Notice that the sum of the in-degrees of all nodes is the number of interconnections of the network. Therefore, this characterization is in line with our prior work on sparsity optimization (c.f. Zavlanos et al. [2008], Julius et al. [2009]).

The smallest set of regulators for each gene is possibly not unique. Consider an example where we can explain the dynamics of the expression activity of Gene X by using the expression activity of Gene Y, or Gene Z. In this case, both {Y} and {Z} are the smallest sets of regulators for X. In this work, we consider the minimality of the set of regulators in the sense of the partial ordering generated by set inclusion. Therefore, a set of regulators G_i^R is considered smaller than another set $G_i^{R\prime}$ if $G_i^R \subset G_i^{R\prime}$. For each gene, we identify the minimal sets of regulators by performing breadth-first search on the lattice structure of the power set of $\{1,\dots,G\}$, which is induced by the set inclusion relation (see Figure 1).

While traversing the lattice, we verify each set whether it qualifies as a set of regulators for the gene of interest. A more detailed description of the verification algorithm is given in Section 4. Although in the worst case, this method requires us to verify all the subsets of $\{1,\dots,G\}$ to identify the minimal sets of regulators, in practice we do not have to search the entire lattice. This is because:

- If a subset $G' \subset \{1,\dots,G\}$ is verified as a set of regulators, any upper bound (i.e. any superset) of G'

is also a set of regulators. This is because the zero function satisfies Assumption 1.

- In most gene regulatory networks found in nature, the in-degree of most genes is small.

4. MAIN RESULT

Given the expression activity data $x_{i,j}$, $1 \leq i \leq G$, $0 \leq j \leq N$, we first compute its (temporal) differential expression $q_{i,j}$, $1 \leq i \leq G$, $0 \leq j \leq N-1$ (see (1)). Then, for each $i \in \{1, \ldots, G\}$ we perform the *ascending sort* operation on the first $N-1$ time points of $x_{i,\cdot}$ to yield $\hat{x}_{i,\cdot}$. That is, there exists a bijection $\sigma_i : \{0, \ldots, N-1\} \rightarrow \{0, \ldots, N-1\}$, such that:

$$x_{i,j} = \hat{x}_{i,\sigma_i(j)}, \tag{4}$$

and for any $j < j'$,

$$\hat{x}_{i,j} \leq \hat{x}_{i,j'}. \tag{5}$$

Thus, simply speaking, the bijection σ_i tells us how the elements of $x_{i,\cdot}$ are permuted by the sort operation.

Now, consider any gene in the network, say, Gene $k \in \{1, \ldots, G\}$. We want to verify the following hypothesis.

Hypothesis 1: A subset $G_k^R \subseteq \{1, \ldots, G\}$ is a set of regulators for Gene k.

Following the discussion in Section 2, we can see that this hypothesis is equivalent to the statement that the data satisfy

$$q_{k,j} = -\lambda_k x_{k,j} + \sum_{m \in G_k^R} f_{k,m}(x_{m,j}), \tag{6}$$

for some $\lambda_k \geq 0$ and CNM functions $f_{k,m}(\cdot)$ for $m \in G_k^R$. Next, we define the variables $\Delta_{m,j}^k$, $m \in G_k^R$, $0 \leq j \leq N-2$, as follows.

$$\Delta_{m,j}^k \triangleq f_{k,m}(\hat{x}_{m,j+1}) - f_{k,m}(\hat{x}_{m,j}). \tag{7}$$

Notice that with these new variables, we have

$$\begin{aligned} f_{k,m}(x_{m,j}) &= f_{k,m}(\hat{x}_{m,\sigma_k(j)}), \\ &= f_{k,m}(\hat{x}_{m,0}) + \sum_{l=0}^{\sigma_k(j)-1} \Delta_{m,l}^k. \end{aligned} \tag{8}$$

In order to verify the hypothesis, we use the following theorem.

Theorem 2. There exist $\lambda_k \geq 0$ and CNM+ functions $f_{k,m}(\cdot)$ for $m \in G_k^R$ such that (6) holds if and only if the following Linear Programming (LP) set of constraints are feasible.

$$\left. \begin{aligned} & q_{k,j} = -\lambda_k x_{k,j} + \sum_{m \in G_k^R} \left(f_{k,m}(\hat{x}_{m,0}) + \sum_{l=0}^{\sigma_k(j)-1} \Delta_{m,l}^k \right), \\ & \forall j \in \{0, \ldots, N-1\}, \end{aligned} \right\} \tag{9}$$

$$\Delta_{m,l}^k \geq 0, \ \forall m \in G_k^R, \ \forall l \in \{0, \ldots, N-2\}, \tag{10}$$

$$\lambda_k \geq 0, \tag{11}$$

$$f_{k,m}(\hat{x}_{m,0}) \geq 0, \ m \in G_k^R \tag{12}$$

$$f_{k,m}(\hat{x}_{m,0}) + \sum_{l=0}^{N-2} \Delta_{m,l}^k \geq 0, \ m \in G_k^R, \tag{13}$$

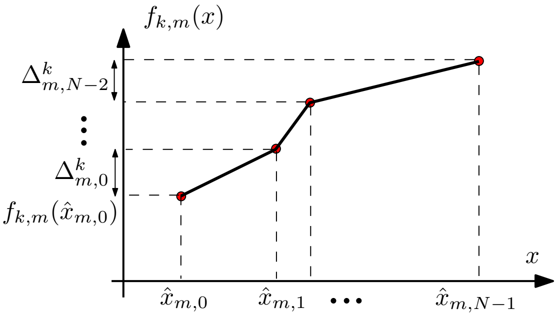

Fig. 2. Constructing a continuous monotonically increasing function $f_{k,m}(x)$ that is compatible with a feasible solution of the LP problem.

with $f_{k,m}(\hat{x}_{m,0})$, $\Delta_{m,l}^k$, and λ_k as the optimization variables.

Proof. (only if) We notice that (9) is obtained by substituting (8) into (6). Suppose that there exist $\lambda_k \geq 0$ and CNM+ functions $f_{k,m}(\cdot)$ for $m \in G_k^R$ such that (6) holds, then the fact that constraints (9) and (11) can be met is trivial. Constraint (10) can be met because of definition (7) and the fact that $f_{k,m}(\cdot)$ is monotonically increasing. Next, notice that the left hand side of (13) is equal to $f_{k,m}(\hat{x}_{m,N-1})$. Therefore, constraints (12) and (13) can be met because of the fact that $f_{k,m}(\cdot)$ is nonnegative.

(if) Suppose that the LP constraints are feasible, for some $f_{k,m}(\hat{x}_{m,0})$, $\Delta_{m,l}^k$, and λ_k. We only need to show the existence of CNM+ functions $f_{k,m}(\cdot)$ for $m \in G_k^R$ that are consistent with these variables, i.e. (8) is satisfied. This can be done quite easily, for example by constructing the functions $f_{k,m}(\cdot)$ for $m \in G_k^R$ as piecewise affine functions connecting the data, as illustrated in Figure 2. ∎

Theorem 2 provides us with a necessary and sufficient condition for a subhypothesis that we want to verify, namely the fact that G_k^R is a set of regulators for Gene k, and that all genes in G_k^R are activators. For a different activator-repressor configuration, it is easy to see that the sign constraint given by (10) can be modified accordingly. Consequently, to verify Hypothesis 1 above, we need to solve $2^{|G_k^R|}$ LP feasibility problems, each corresponding to an activator-repressor partition of G_k^R.

In summary, the method that we develop in this paper is presented in Algorithm 1.

5. NUMERICAL IMPLEMENTATION AND EXAMPLE

In the numerical implementation, we modify the algorithm above as follows. Instead of checking the feasibility of the LP problem in Theorem 2, we minimize the magnitude of the slack variables to obtain a feasible solution. That is, we introduce the slack variables $\varepsilon_{m,j}^{\Delta}$, $m \in G_k^R$, $0 \leq j \leq N-2$, ε_m^f, $m \in G_k^R$ and formulate a Linear Quadratic (LQ) programming problem.

Algorithm 1 Computation of the sparsest gene network based on gene expression data time-series

```
Require: Time-series data of gene expression activities $x_{i,j}$.
Compute the time differential expression data $q_{i,j}$.
for all $k \in \mathcal{G} \triangleq \{1, \ldots, G\}$ do
    Compute the lattice of the subsets of $\mathcal{G}$ as in Figure 1. Label every subset with 'unverified'.
    repeat
        Take a subset of $G_k^R \subset \mathcal{G}$ with label 'unverified'.
        for all possible activator-repressor partitioning of $G_k^R$ do
            Use Theorem 2 to verify whether $G_k^R$ is a set of regulators for Gene $k$.
        end for
        if $G_k^R$ is a set of regulators for Gene $k$ then
            Label $G_k^R$ and all of its upper bounds in the lattice with 'pass'.
        else
            Label $G_k^R$ with 'fail'.
        end if
    until all subsets are labeled with 'pass' or 'fail'.
    The possible sets of regulators for Gene $k$ are all subsets labelled with 'pass'. The minimal sets of regulators are the minimal elements (in the sense of set inclusion) of this set.
end for
```

$$\min \|\varepsilon\|_F \,, \text{ subject to}$$

$$\left.\begin{array}{l} q_{k,j} = -\lambda_k x_{k,j} + \displaystyle\sum_{m \in G_k^R} \left(f_{k,m}(\hat{x}_{m,0}) + \sum_{l=0}^{\sigma_k(j)-1} \Delta_{m,l}^k \right), \\ \forall j \in \{0, \ldots, N-1\}, \end{array}\right\}$$

$$\Delta_{m,l}^k + \varepsilon_{m,l}^\Delta \geq 0, \ \forall m \in G_k^R, \ \forall l \in \{0, \ldots, N-2\},$$

$$\lambda_k \geq 0,$$

$$f_{k,m}(\hat{x}_{m,0}) + \varepsilon_m^f \geq 0, \ m \in G_k^R,$$

$$\left(f_{k,m}(\hat{x}_{m,0}) + \varepsilon_m^f\right) + \sum_{l=0}^{N-2} \left(\Delta_{m,l}^k + \varepsilon_{m,l}^\Delta\right) \geq 0, \ m \in G_k^R,$$

with $f_{k,m}(\hat{x}_{m,0})$, $\Delta_{m,l}^k$, $\varepsilon_{m,l}^\Delta$, ε_m^f and λ_k as the optimization variables. The symbol $\varepsilon := \begin{bmatrix} \varepsilon^f \\ \varepsilon^\Delta \end{bmatrix}$ and $\|\cdot\|_F$ denotes Frobenius norm. The optimum of the LQ problem can therefore be considered as the distance of $f_{k,\cdot}(\hat{x}_{\cdot,0})$ and $\Delta_{\cdot,\cdot}^k$ from being a feasible solution to the LP constraints. For example, if the original LP constraints in Theorem 2 are feasible, then we can set $\varepsilon = 0$, and thus the LQ problem above will attain its global minimum at 0. On the other hand, the further $f_{k,\cdot}(\hat{x}_{\cdot,0})$ and $\Delta_{\cdot,\cdot}^k$ are from being a feasible solution, the higher the optimal solution for the LQ problem.

5.1 Numerical Example

As a proof of concept, we test our algorithm on an *in silico* data set. The data set is obtained from a mathematical model of the synthetic gene network known as the *repressilator*. The repressilator is a synthetic oscillator network that was originally conceived and constructed by Elowitz and Leibler (see Elowitz and Leibler [2000]). The network consists of three genes in a repressive cycle.

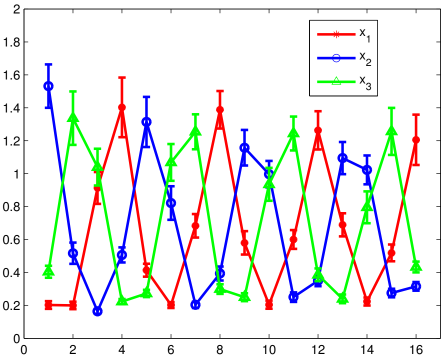

Fig. 3. The time-series data of gene expression activities.

A mathematical model of the dynamics of the repressilator that includes both the transcription regulation and translation dynamics is given as follows (see e.g. Alon [2007]).

$$\left.\begin{array}{ll} \dfrac{dx_1}{dt} = \dfrac{100}{1+8y_2^3} - 50x_1, & \dfrac{dy_1}{dt} = x_1 - y_1, \\ \dfrac{dx_2}{dt} = \dfrac{100}{1+8y_3^3} - 50x_2, & \dfrac{dy_2}{dt} = x_2 - y_2, \\ \dfrac{dx_3}{dt} = \dfrac{100}{1+8y_1^3} - 50x_3, & \dfrac{dy_3}{dt} = x_3 - y_3. \end{array}\right\} \quad (14)$$

Here, the symbols $x_{1,2,3}$ denote the concentrations of the mRNA transcripts of Genes 1, 2, and 3, respectively. The symbols $y_{1,2,3}$ denote the protein concentrations of the respective genes. For simplicity, the parameters of the model are chosen symmetrically. In choosing these parameters, we follow the biological/experimental knowledge that the mRNA decay is much faster than protein decay (see Alon [2007]).

The ODE (14) represents the topology where Gene 1 is repressed by Gene 2, Gene 2 is repressed by Gene 3, and Gene 3 is repressed by Gene 1.

Using the standard forward Euler method to solve the ODE, we obtain the 3D trajectory of the mRNA transcript concentrations. These trajectories are then sampled to generate a time-series of gene expression activity data. To model measurement noise, we add uncorrelated Gaussian noise on each sample. A time plot showing these samples with error bars indicating their respective standard deviations are shown in Figure 3.

We apply the algorithm presented in the previous section on this data set. To test the robustness of the algorithm against noise, we randomly generate 20 data sets and apply the algorithm on each of data sets. We implement the algorithm in MATLAB, with the `cvx` toolbox used for solving the LQ problem (see Boyd and Grant [2005]). The calculation is run on a standard laptop computer (Intel Core2 Duo P8600 2.4GHz with 3GB RAM). Each data set takes less than 5 seconds to process. The solutions to the LQ problem above for various regulator sets for Gene 1 is shown in Table 1.

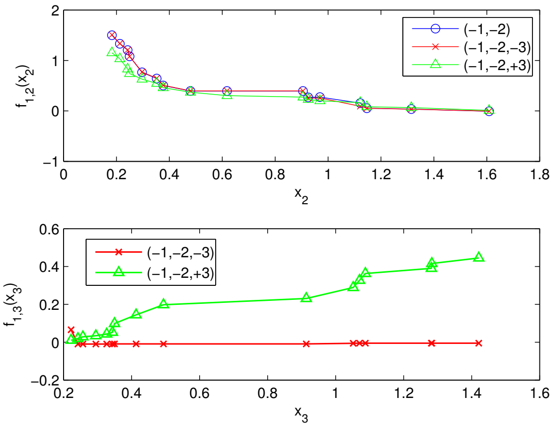

Fig. 4. (Top) The regulatory effect of Gene 2 on Gene 1, computed for one of the 20 datasets. (Bottom) The regulatory effect of Gene 3 on Gene 1, computed for the same dataset.

Reg. Set	$\|\varepsilon\|_F$	Reg. Set	$\|\varepsilon\|_F$
(-1)	1.77 ± 0.22	(+2,+3)	0.15 ± 0.048
(-2)	0.24 ± 0.11	(+2,-3)	0.31 ± 0.039
(+2)	0.70 ± 0.12	(-2,+3)	0.030 ± 0.037
(-3)	0.88 ± 0.18	(-2,-3)	0.074 ± 0.047
(+3)	0.55 ± 0.20	(-1,+2,+3)	0.091 ± 0.032
(-1,+2)	0.36 ± 0.040	(-1,+2,-3)	0.23 ± 0.022
(-1,-2)	0.057 ± 0.044	(-1,-2,+3)	0.012 ± 0.022
(-1,+3)	0.18 ± 0.058	(-1,-2,-3)	0.025 ± 0.023
(-1,-3)	0.38 ± 0.052		

Table 1. The solutions to the LQ problem for various regulator sets of Gene 1.

In the table above, the notation for the regulator sets can be explained as follows. The set (-1) means the Gene 1 acts as a repressor. The set (-1,-2,+3) means Gene 1 and 2 act as repressors, while Gene 3 acts as an activator, and so on. Notice that we do not verify any regulator set with Gene 1 itself as an activator. The reasoning behind it is explained in the next section.

Some observation from the data:

- As is generally true with any noisy data, larger regulator sets allow for better fit of the data. This is due to overfitting, where the algorithm uses the extra degrees of freedom to explain the noise.
- Among the regulator sets with one member, the algorithm correctly pick Gene 2 as the repressor of Gene 1. The next best fit (by a factor of more than 2) is Gene 3 as the activator of Gene 1. This is because Gene 3 is an indirect activator of Gene 1 (through Gene 2). However, since we are interested in direct regulation relationships, the algorithm correctly discounts this hypothesis. See Figure 4 (top) where we plot the regulatory effect of Gene 2 on Gene 1 (represented by x_2 vs $f_{1,2}(x_2)$), in three different regulator sets, computed for one of the 20 data sets. The function $f_{1,2}(x_2)$ in (-1,-2) and (-1,-2,-3) are very close. It drops a little in (-1,-2,+3), which is allowed by the presence of indirect regulation by Gene 3, as seen in Figure 4 (bottom).

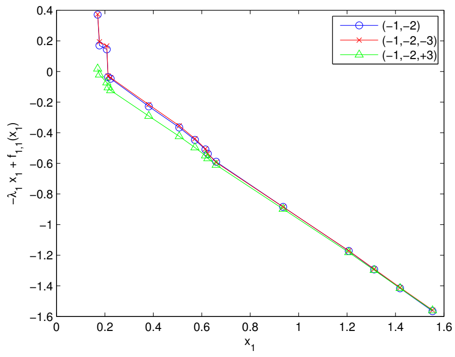

Fig. 5. The regulation of Gene 1 by itself, computed from one of the 20 datasets.

- The top 4 overall best fits (by a factor of more than 2 compared to the rest) share the common element, i.e. Gene 2 as repressor. Three of them share the element of Gene 1 as a repressor. This is probably due to the fact that the linear decay assumption is not perfectly accurate. Therefore, the algorithm uses Gene 1 as an autorepressor to fix the decay model. See Figure 5, where we plot the effect of Gene 1 on its own dynamics (i.e. x_1 vs $-\lambda_1 x_1 + f_{1,1}(x_1)$). It can be seen that from the three different regulator sets, the algorithm predicts that the effect is mostly linear.

6. IRREFUTABLE REGULATION HYPOTHESES

As discussed earlier, our method is based on refuting regulation hypotheses for each gene in the network. That is, we reject the hypothesis about a given regulator set, if the data set cannot be fit in the model (2) while satisfying all assumptions about the regulation functions f. In this section, we discuss irrefutable regulation hypotheses, which are conditions where: (i) the regulator set is such that it cannot be refuted, regardless of the data, or (ii) the expression activity patterns of some genes are such that they cannot be refuted as regulators of any gene regardless of its expression data.

6.1 Autoactivation

The hypothesis that a gene acts as its own activator, in practice, cannot be refuted. This is because of the following lemma.

Lemma 3. Given a finite data set consisting of gene expression data x_n and time-differential expression data q_n, $n \in \{1, \ldots, N\}$ satisfying $x_n \geq 0$, and $x_i \neq x_j$ if $i \neq j$. There exist $\lambda \geq 0$ and f a CNM+ function such that

$$q_n = -\lambda x_n + f(x_n), \; n \in \{1, \ldots, N\}. \tag{15}$$

Proof. Without any loss of generality, we assume that $(x_n)_{n=1,\ldots,N}$ is an increasing sequence of numbers [2]. Pick any $\tilde{\lambda}$ such that

$$\tilde{\lambda} \geq \max_{1 \leq n \leq N} \frac{q_n - q_{n-1}}{x_n - x_{n-1}}, \tag{16}$$

where we define $x_0 = q_0 := 0$.We claim that $q_n + \tilde{\lambda} x_n$ is an increasing sequence. To see this, we observe that

$$q_n + \tilde{\lambda} x_n - q_{n-1} - \tilde{\lambda} x_{n-1} = q_n - q_{n-1} + \tilde{\lambda}\,(x_n - x_{n-1}) \geq 0.$$

[2] Otherwise, we can always sort the data beforehand.

Therefore, using the same argument as in the proof of Theorem 2, we can construct a CNM+ function f such that (15) holds. ■

The two assumptions that we impose in this lemma, i.e. the nonnegativity and uniqueness of the expression data always hold in practice. The nonnegativity property obviously comes from the fact that mRNA transcript concentrations cannot be a negative number. The uniqueness property always holds in practice, because in noisy measurements the probability of two different measurements yield exactly the same number is 0.

6.2 Monotonic Time Series

When the time series of the expression activities of two genes are monotonic, they can be used as universal regulators. That is, they cannot be refuted as regulators of any gene, regardless of the expression data. This is proved in the following lemma.

Lemma 4. Given a finite data set consisting of gene expression data of two genes $x_{1,n}$ and $x_{2,n}$, $n \in \{1, \ldots, N\}$. We assume that:

- $x_{1,n} \geq 0$, $x_{2,n} \geq 0$,
- $x_{1,i} \neq x_{1,j}$, $x_{2,i} \neq x_{2,j}$ if $i \neq j$,
- $x_{1,n}$ and $x_{2,n}$ are both monotonically increasing sequence.

For any gene in the network, suppose that we have the expression data x_n and the corresponding time-differential expression data q_n, $n \in \{1, \ldots, N\}$. There exist $\lambda \geq 0$, f_1 a CNM+ function, and f_2 a CNM- function such that

$$q_n = -\lambda x_n + f_1(x_{1,n}) + f_2(x_{2,n}), \ n \in \{1, \ldots, N\}.$$

Proof. Pick any $\lambda \geq 0$. We need to find appropriate f_1 and f_2 such that

$$q_n - \lambda x_n = f_1(x_{1,n}) + f_2(x_{2,n}).$$

Then, we use the fact that any sequence can be written as the sum of a monotonically increasing and a monotonically decreasing sequence to establish the existence of f_1 and f_2. ■

Notice that this lemma specifically assumes that the expression activity of Gene 1 and Gene 2 are monotonically increasing. However, we can see that if they were both monotonically decreasing, the same result would still hold. This is because we can pre-sort the data to make both $x_{1,n}$ and $x_{2,n}$ monotonically increasing.

If the expression activity of Gene 1 is monotonically increasing, and Gene 2 is monotonically decreasing, we can require that both f_1 and f_2 are of the same type (i.e. CNM+ or CNM- functions). In this case, the same result would also hold. In fact, the result above can be generalized by stating that any sequence of expression activities of two genes, $x_{1,n}$ and $x_{2.n}$, can be used as universal regulators if the sequence of ordered pairs $(x_{1,n}, x_{2,n})$ can be permuted such that the sequence of the first elements and the sequence of the second elements are both monotonic.

Discussion The assumptions of CNM functions are very general. However, the assumption that the regulatory functions are linear combinations of CNM functions is somewhat restrictive. In future work, we plan on generalizing this regulation assumption to the case of general multivariate CNM functions.

ACKNOWLEDGEMENTS

This work was partially supported by NSF grant number 1000284 and NIH R01 grant number DE15989. The authors would like to thank Dr. Salomon Amar and Niraj Trivedi (both at Boston University) for the constructive discussions that lead to this paper.

REFERENCES

U. Alon. *An Introduction to Systems Biology.* Chapman and Hall, 2007.

E. August and A. Papachristodoulou. Efficient, sparse biological network determination. *BMC Systems Biology*, 3(25), 2009.

R. Bonneau, D. J. Reiss, P. Shannon, M. Facciotti, L. Hood, N.S. Baliga, and V. Thorsson. The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets *de novo. Genome Biology*, 7:R36, 2006.

S. Boyd and M. C. Grant. cvx – MATLAB software for disciplined convex programming, 2005. http://www.stanford.edu/~boyd/cvx/.

S. Drulhe, G. Ferrari-Trecate, and H. de Jong. The switching threshold reconstruction problem for piecewise affine models of genetic regulatory networks. *IEEE Trans. Automatic Control*, 53(1):153–165, 2008.

M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767): 335–338, 2000.

J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner. Large-scale mapping and validation of *escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, 5(1):e8, 2007.

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

A. A. Julius, M. M. Zavlanos, S. P. Boyd, and G. J. Pappas. Genetic network identification using convex programming. *IET Systems Biology*, 3(3):155–166, 2009.

R. Porreca, E. Cinquemani, J. Lygeros, and G. Ferrari-Trecate. Identification of genetic network dynamics with unate structure. *Bioinformatics*, 26(9):123–1245, 2010.

R. Porreca, S. Drulhe, H. de Jong, and G. Ferrari-Trecate. Structural identification of piecewise-linear models of genetic regulatory networks. *Journal of Computational Biology*, 15:1365–1380, 2008.

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

M. K. S. Yeung, J. Tegner, and J. J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. of the National Academy of Science*, 99(9):6163–6168, 2002.

M. M. Zavlanos, A. A. Julius, S. Boyd, and G. J. Pappas. Identification of stable genetic networks using convex programming. In *Proc. American Control Conference*, Seattle, USA, 2008.