International Journal of Biological, Life and Agricultural Sciences
ISSN: 2415-6612
Vol:12, No:10, 2018

Improved Predictive Models for the IRMA Network Using Nonlinear Optimisation

Vishwesh Kulkarni, Nikhil Bellarykar

***Abstract*—**Cellular complexity stems from the interactions among thousands of different molecular species. Thanks to the emerging fields of systems and synthetic biology, scientists are beginning to unravel these regulatory, signaling, and metabolic interactions and to understand their coordinated action. Reverse engineering of biological networks has has several benefits but a poor quality of data combined with the difficulty in reproducing it limits the applicability of these methods. A few years back, many of the commonly used predictive algorithms were tested on a network constructed in the yeast *Saccharomyces cerevisiae* (S. cerevisiae) to resolve this issue. The network was a synthetic network of five genes regulating each other for the so-called *in vivo reverse-engineering and modeling assessment* (IRMA). The network was constructed in S. cereviase since it is a simple and well characterized organism. The synthetic network included a variety of regulatory interactions, thus capturing the behaviour of larger eukaryotic gene networks on a smaller scale. We derive a new set of algorithms by solving a nonlinear optimization problem and show how these algorithms outperform other algorithms on these datasets.

***Keywords*—**Synthetic gene network, network identification, nonlinear modeling, optimization.

I. Introduction

IN principle, interactions among genes, when unknown, can be identified from transcriptomic data using reverse-engineering methods. Typically, the data consist of measurements at steady state after multiple perturbations (i.e., gene over-expression, knockdown, or drug treatment) or at multiple time points after one perturbation (i.e., time series data). Successful applications of these approaches have recently been demonstrated in bacteria, yeast, and mammalian systems [1]-[4]. The main difficulty encountered by these reverse engineering methods is due to the poor and often non-reproducible quality of the data and the scarcity of such datasets. In [1], an interesting experiment was performed and some of the most successful such predictive algorithms were tested on a network constructed in the yeast *Saccharomyces cerevisiae* (S. cerevisiae). The network, shown in Fig. 1 and commented upon in [4], is a synthetic network of five genes regulating each other for the so-called *in vivo reverse-engineering and modeling assessment* (IRMA). In this paper, we show how nonlinear modeling can improve upon the results presented in [1].

II. Problem Formulation

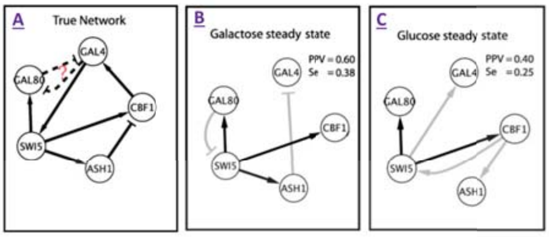


Fig. 1 (A) The correct IRMA network as given and (B)-(C) the NIR results given [1]

We first present an overview of the gene network identification problem solved in [1]. The variables of interest are protein concentrations, which are assumed to be proportional to the mRNA abundances of five genes ($[CBF1]=x_1$; $[GAL4]=x_2$; $[SWI5]=x_3$; $[GAL80]=x_4$; $[ASH1]=x_5$) The simulated network dynamics are given by

$$\begin{aligned}
\frac{dx_1}{dt} &= \alpha_1 - d_1x_1 + v_1\frac{x_3^{h_1}(t-\tau)}{\left(k_1^{h_1}+x_3^{h_1}(t-\tau)\right)}\frac{1}{\left(1+\frac{x_5^{h_2}}{k_2^{h_2}}\right)}+u_1\\
\frac{dx_2}{dt} &= \alpha_2 - (d_2-\Delta(\beta_1))x_2 + v_2\frac{x_1^{h_3}}{k_3^{h_3}+x_1^{h_3}}+u_2\\
\frac{dx_3}{dt} &= \alpha_3 - d_3x_3 + v_3\frac{x_2^{h_4}}{k_4^{h_4}+x_2^{h_4}}\frac{1}{\left(1+\frac{x_4^4}{\gamma^4}\right)}+u_3 \qquad (1)\\
\frac{dx_4}{dt} &= \alpha_4 - (d_4-\Delta(\beta_2))x_4 + v_4\frac{x_3^{h_5}}{k_5^{h_5}+x_3^{h_5}}+u_4\\
\frac{dx_5}{dt} &= \alpha_5 - d_5x_5 + v_5\frac{x_3^{h_6}}{k_6^{h_6}+x_3^{h_6}}+u_5
\end{aligned}$$

The functions $\Delta(\beta)$ are pulse functions with magnitude β between 0 and 10 seconds, and magnitude 0 otherwise. The functions $\Delta(\beta)$ are used to simulate the effect of an initial washing step (see bottom of page 2 of the supplementary material). The signals u_i are inputs that will be discussed later. The other parameters are shown in Table I towards the end of this manuscript and are for growth in a galactose medium or glucose medium. The variables that change between the two media are noted. For later use, this system is represented in compact form as

$$\dot{x} = f(x) + u \tag{2}$$

The goal is to determine the topology of this interconnected system, that is, which variables appear on the right hand side for each differential equation. The experimental data consists of different types of perturbations with measurements both at steady state and as a function of time.

Fig. 3 of [1] shows time dependent data after switching between growing media. "Switch on" data moves from glucose to galactose, and "Switch off" moves from galactose to glucose. To replicate the switch on data, we first simulate

Nikhil Bellarykar is with Tata Consultancy Services, Pune, India. Vishwesh Kulkarni is with the School of Engineering, University of Warwick, Coventry, CV4 7AL, UK (e-mail: nikhil.bellarykar@tcs.com, V.Kulkarni@warwick.ac.uk).

(1) to steady state with parameters for glucose, and record the final states. We then perform a simulation using galactose parameters, but with initial condition as the final state for the glucose simulation. A similar simulation is done to switch to glucose. The results are shown in Fig. 2, and are close to that of Fig. 3 of [1].

Fig. 4 of [1] shows steady state measurements after overexpression experiments in both growth media. (Experimental data is on top, data simulated using the dynamical equations above is below). In an overexpression experiment, a factor is introduced which increases the transcription rate of a particular gene. This is modeled in the dynamical equations by the application of a non-zero u_i for a particular gene, with the remaining $u_j = 0$, $i \neq j$. We attempt to replicate this data here. In Fig. 3 steady state expression in both galactose and glucose is plotted for no perturbation, along with perturbation for each gene. The input u_i is selected so that the results are close to that of Fig. 4 in [1], but the magnitude of this input is not recorded.

If the system is operated over a restricted region, and the function f is sufficiently smooth, then (2) can be well approximated by a linear dynamical equation. Let x_0 be the equilibrium with $u = 0$, and let $x' = x - x_0$. Then a linear approximation takes the form

$$\frac{dx'}{dt} = Ax' + u \tag{3}$$

where $A \in \mathbb{R}^{n \times n}$ is a matrix with non-zero diagonal elements, and a non-zero entry at element (i, j) when $j \in \mathscr{N}_i$. This model is the basis for a variety of recovery methods, including NIR [5] and TSNI [2], which are tested in [1].

We will be interested here in the case when the state velocity is estimable. In this case, data from both steady state and dynamic experiments needs to be written a the common framework. Suppose that there are N measurements at distinct times and/or experiments with the index k indexing the measurements. Then, modulo measurement noise, the experimental data can be described by the relationship

$$Z - U = XA^T \tag{4}$$

where $Z \in \mathbb{R}^{n \times N}$ is given by

$$Z = \begin{bmatrix} \left(\frac{dx}{dt}\right)_1 & \cdots & \left(\frac{dx}{dt}\right)_N^T \end{bmatrix}^T$$

(where $\left(\frac{dx}{dt}\right)_k = 0$ in the case of steady state experiments), $U \in \mathbb{R}^{n \times N}$ is a matrix with sparse support, given by

$$U = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}^T$$

where the location of the non-zero elements is known, although the magnitude may be unknown, and $X \in \mathbb{R}^{n \times N}$ a matrix containing state measurements of the form

$$X = \begin{bmatrix} x'_1 & x'_2 & \cdots & x'_N \end{bmatrix}^T$$

The recovery of the network interconnectivity can be achieved by solving for A, and identifying the elements that are non-zero. From the form of (4) it is clear that each row of A is involved in an independent equation. Let $Z^{i\downarrow}$ be the ith column of Z, and similarly for U and A^T. Then

$$Z^{i\downarrow} - U^{i\downarrow} = X(A^T)^{i\downarrow}$$

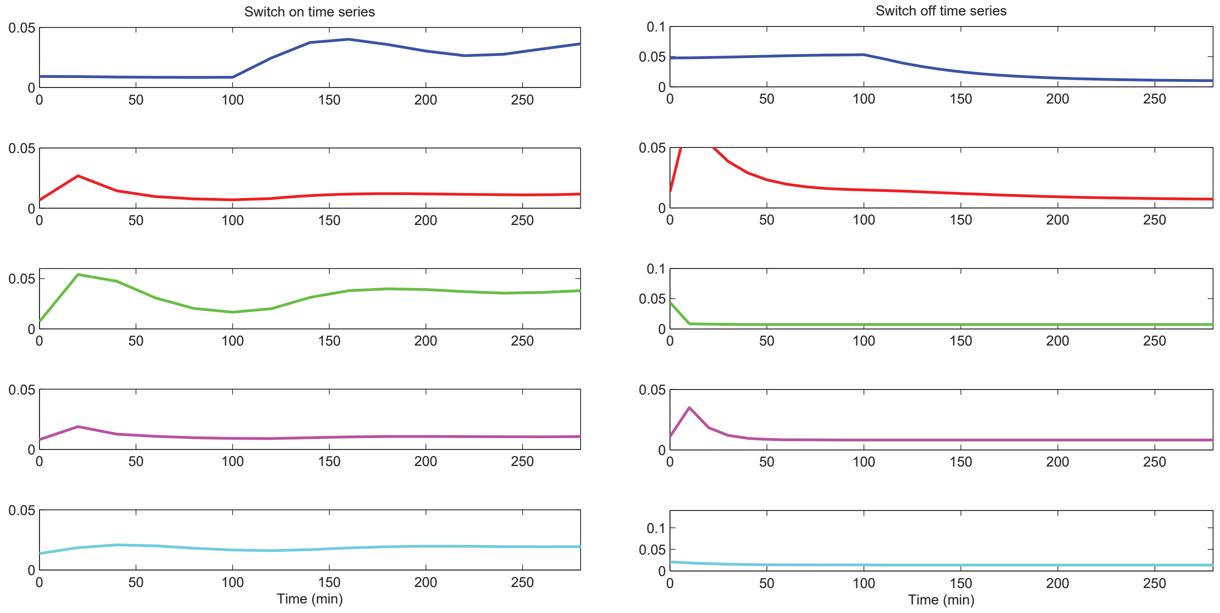
Algorithm 1 The NIR algorithm

```
input: measurements $U^{i\downarrow}$, $X$, max elements $k$
for $i = 1$ to $\binom{n}{k}$ do
    a. $\Pi =$ set of $k$ elements of $\{1, 2, \cdots, n\}$
    b. estimate: $\alpha_i = (X^{\Pi\downarrow})^{\dagger} U^{i\downarrow}$
    c. residual magnitude: $r_i = \|U^i - X^{\Pi\downarrow}\alpha_i\|_2$
end for
output: $\alpha_i$ with minimum $r_i$
```

By solving this equation for $(A^T)^{i\downarrow}$, we can obtain an estimate for the ith row of A. The following methods have been proposed:

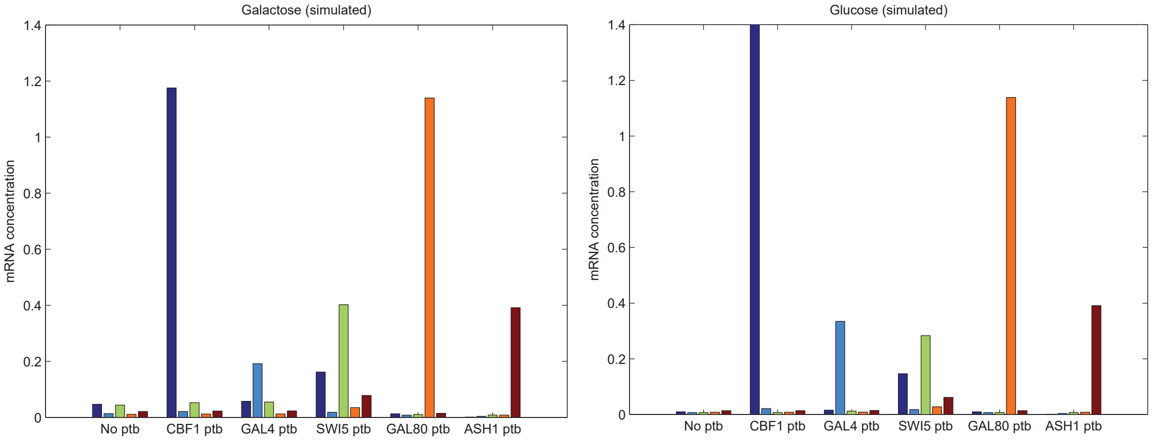
- The NIR algorithm [6] was proposed for steady state data ($Z^{i\downarrow} = 0$) and solves for $A^{i\downarrow}$ by exhaustive search of different combinations of non-zero elements of $A^{i\downarrow}$. Specifically, an upper limit on the number of non-zero elements, denoted k, is selected, and Algorithm 1 is implemented. Note that in many cases the magnitude of $U^{i\downarrow}$ is unknown. However, since only one element is non-zero, the magnitude can be arbitrarily set to 1, which simply scales the estimate by the true magnitude. If more than one element were non-zero, this would not be feasible.
- A more advanced version of the NIR algorithm has been proposed in [7], which will be called the ℓ_1 method. In this case, an ℓ_1 regularization term is utilized instead of exhaustive search, and they also add linear constraints on the elements of A denoted by inclusion in a set S. Specifically, they recursively solve the optimization problem
$$\begin{array}{rl} \min_A & t\sum_{i,j=1}^{n} w_{ij}|a_{ij}| + (1-t)\varepsilon \\ \text{subject to} & \|XA^T + U\|_1 \leq \varepsilon, \quad A \in S \end{array}$$
where a_{ij} is the i, jth element of A, $\|M\|_1$ is the absolute sum of the matrix M, w_{ij} is a weight chosen using the prior estimate of A
$$w_{ij} = \frac{\delta}{\delta + |a_{ij}|}$$
and t and δ are user selected weights. In addition, they suggest two methods for ensuring the stability of A, either adding constraints using the Gershorin bound, or by adding constraints based on a Lyapunov inequality. Note that while the objective function includes all elements of A, it is really the stability constraints that require this problem to be solved all at once, rather than a row at a time as with the NIR algorithm. Just as with the NIR algorithm, if U is diagonal, then the magnitude can be arbitrarily set to 1, which simply scales the rows of A accordingly.
- The TSNI algorithm [2] is the transient counterpart to the NIR algorithm. As originally stated in [2], the linear model is converted to a sampled data system, so the system of equations to be solve is
$$X_{k+1} = X_k A_d^T + U B_d^T$$
where $A_d = e^{AT_s}$ and $B_d = \int_0^{T_s} e^{At} dt$. They also simply use a pseduo inverse to solve for B_d and A_d, without a sparsification step.

Note that all of the methods above implicitly assume a linear model. The the approach detailed in the next section,



(a) (b)

Fig. 2 *Transient expression data – (a) after switch between glucose and galactose; (b) after switch between galactose and glucose*



(a) (b)

Fig. 3 Nominal steady state expression data, along with steady state data after overexpression for each gene (ptb stands for perturbation) – (a) galactose media; (b) glucose media

we will try to accommodate is the fact that the true model is nonlinear.

III. Methodology and Assumptions

The motivation for our method is based on the following assumptions

A1 The response of each gene in the network is regulated by its own expression level and that of a small number of other genes. For gene i, the set of regulating genes is $\mathcal{N}_i$.

A2 The gene network data is generated by a set of ordinary differential equations where the state velocity function has a constant term, a linear self-regulation term, and a nonlinear term that is a sum of univariate functions.

$$\dot{x}_i = \alpha_i - d_i x_i + \sum_{j\in\mathcal{N}_i} f_{ij}(x_j) + u_i$$

A3 Each nonlinear term f_{ij} can be well approximated by a piecewise-linear function

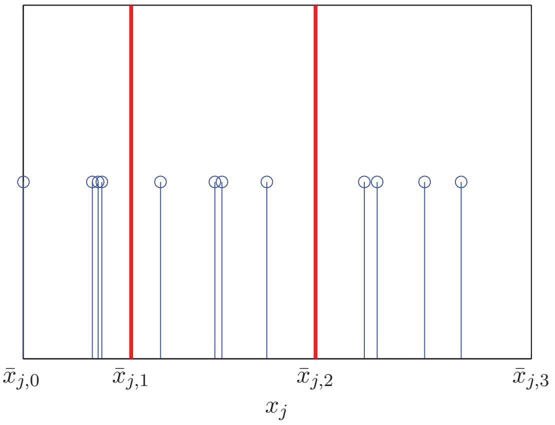


Fig. 4 Example distribution of 50 data points for one node after 10 experiments

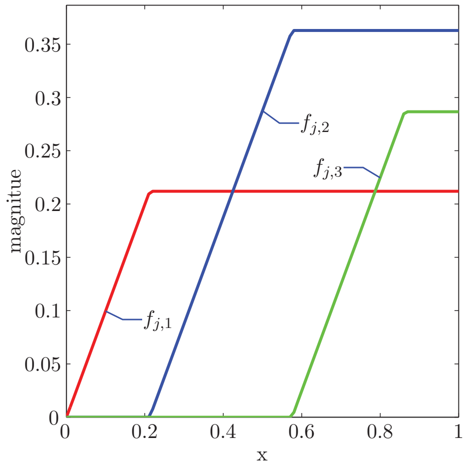


Fig. 5 Examples of our saturation functions

A4 The system is well damped, so that the self-regulation term dominates the dynamics

A5 The experimental data is steady state expression levels with a perturbation input u_i that is non-zero for only one i.

Note that the yeast system described in the first section satisfies these assumptions.

Because of assumptions A2 and A3, we will utilize a small set of piecewise linear basis functions to represent each f_{ij}. Specifically, define $s_{i,j}$ to be the set of saturation functions

$$s_{i,j}(x) = \begin{cases} 0 & x \geq \bar{x}_{i,j-1} \\ x - \bar{x}_{i,j-1} & \bar{x}_{i,j-1} \leq x \leq \bar{x}_{i,j} \\ \bar{x}_{i,j} - \bar{x}_{i,j-1} & x \leq \bar{x}_{i,j} \end{cases}$$

The points of division, $\bar{x}_{i,j}$, are chosen to equalize the saturation based on the observed distribution of the observed data, x_i. For example, suppose that $x_i(k)$ has the distribution shown in Fig. 4, where each stem represents the location of one data point, and we choose $p = 3$. The red lines indicate division points such that one third of the data lies in each region. Based on these divisions, the resulting saturation functions are shown in Fig. 5.

Let $c = \begin{bmatrix}\alpha_1 & \cdots & \alpha_n\end{bmatrix}'$ and $d = \begin{bmatrix}d_1 & \cdots & d_n\end{bmatrix}'$. Based on the assumption set A, the following set of equations will approximate the gene network dynamics

$$\frac{dx'}{dt} = c + \mathrm{diag}(d)x' + A_e x_e + u \tag{5}$$

where $A_e \in \mathbb{R}^{n \times np}$ and

$$x_e = \begin{bmatrix} s_{1,1}(x_1') & \cdots & s_{1,p}(x_1') & s_{2,1}(x_2') & \cdots & s_{n,p}(x_n') \end{bmatrix}^T. \tag{6}$$

Grouping the observed data as in the previous section, the system of equations then becomes

$$Z - U = X_e A_e^T + \mathbf{1}c' + X\mathrm{diag}(d)$$

where $\mathbf{1}$ is a vector of 1s, and $X_e \in \mathbb{R}^{(np) \times N}$ a matrix containing state measurements of the form

$$X_e = \begin{bmatrix} x_{e,1} & x_{e,2} & \cdots & x_{e,N} \end{bmatrix}^T.$$

Our recovery algorithm is based on repeated solution of an optimization problem. To state the optimization problem, we make the following definitions

- Since each state x is associated with multiple elements of x_e, we will want to define a regularization term that will penalize the elements as a group. Define Ω_i to be the indices of x_e that are a mapping of the variable x_i. For example, for x_e defined in (6), $\Omega_1 = \{1, 2, \cdots, p\}$, $\Omega_2 = \{p+1, p+2, \cdots, 2p\}$, etc.
- Define $(A_e^T)^{i\downarrow}_{\Omega_j}$ to be the elements of $(A_e^T)^{i\downarrow}$ with indices in Ω_j.
- As discussed above, the magnitude of U is unknown, but the location of non-zero elements is known. Define

$$\mathbb{U}_i = \{u | u \text{ satisfies appropriate sparsity pattern for } U^{i\downarrow}\}$$

Our recovery algorithm can then be written as follows. For $i = 1, \cdots, n$, recursively solve

$$\begin{array}{cc} \min_{(A_e^T)^{i\downarrow}, U^{i\downarrow}, c_i, d_i} & t\sum_{j=1}^{n} w_k \|(A_e^T)^{i\downarrow}_{\Omega_j}\|_2 + (1-t)\varepsilon \\ \text{subject to} & \|X_e(A_e^T)^{i\downarrow} + \mathbf{1}c_i + X^{i\downarrow}d_i + U^{i\downarrow}\|_2 \leq \varepsilon, \\ & U^{i\downarrow} \in \mathbb{U}_i \quad (A_e^T)^{i\downarrow}_{\Omega_i} = 0 \quad d_i = -1 \end{array}$$

where the weight

$$w_k = \frac{\delta}{\delta + \|(\bar{A}_e^T)^{i\downarrow}_{\Omega_k}\|_2}$$

is calculated from the priori estimate of $(A_e^T)^{i\downarrow}_{\Omega_k}$, with the normalization

$$(\bar{A}_e^T)^{i\downarrow} = \frac{(A_e^T)^{i\downarrow}}{\|(A_e^T)^{i\downarrow}\|_2}.$$

Note that we have assumed that the linear self regularization term is dominant, so that no nonlinear basis elements associated with x_i are needed, and thus we can take $(A_e^T)^{i\downarrow}_{\Omega_i} = 0$. Also, since we are solving for $U^{i\downarrow}$ we normalize the variables by setting $d_i = -1$, which assumes a stable self-regularization term.

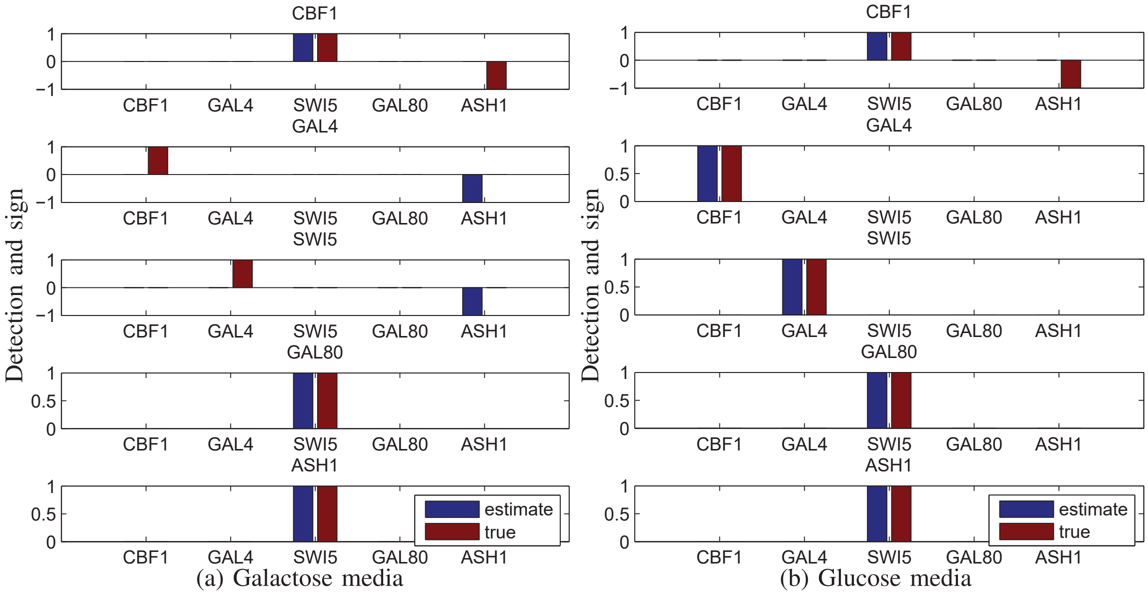
IV. Results

In this section we compare the results on steady state data for the NIR, ℓ_1, and our approach. In [1], the NIR approach was also tested, with the results shown in Fig. 5 of that paper. As was done in Cantone, we search all combinations of 2 input genes (including self loops). Since the best fit pair always included a self loop, this method essentially was limited to finding one other input gene.

One irregularity is what [1] considered to be the correct network. Their correct network is reproduced in our Fig. 1 (a). The red question mark has been added to denote the connections that do not exist the system of equations (1). It could be that this is why these connections are dotted, but this is not discussed in the figure caption.

Unfortunately, our testing of the NIR method was not able to replicate the results of [1]. The results from using the NIR algorithm in the [1]. paper are repeated in our Fig. 1, while the results of using the NIR algorithm with the data generated ourselves (and plotted in our Fig. 3) is shown in Fig. 6. While the Galactose media results are somewhat similar, we show an inhibitory input from ASH1 to SWI5, while they show an inhibitory input from GAL80 to SWI5. The results for Glucose media are quite different, with our results much closer to the true network. Essentially only the inhibitory input from ASH1 to CBF1 is missing.

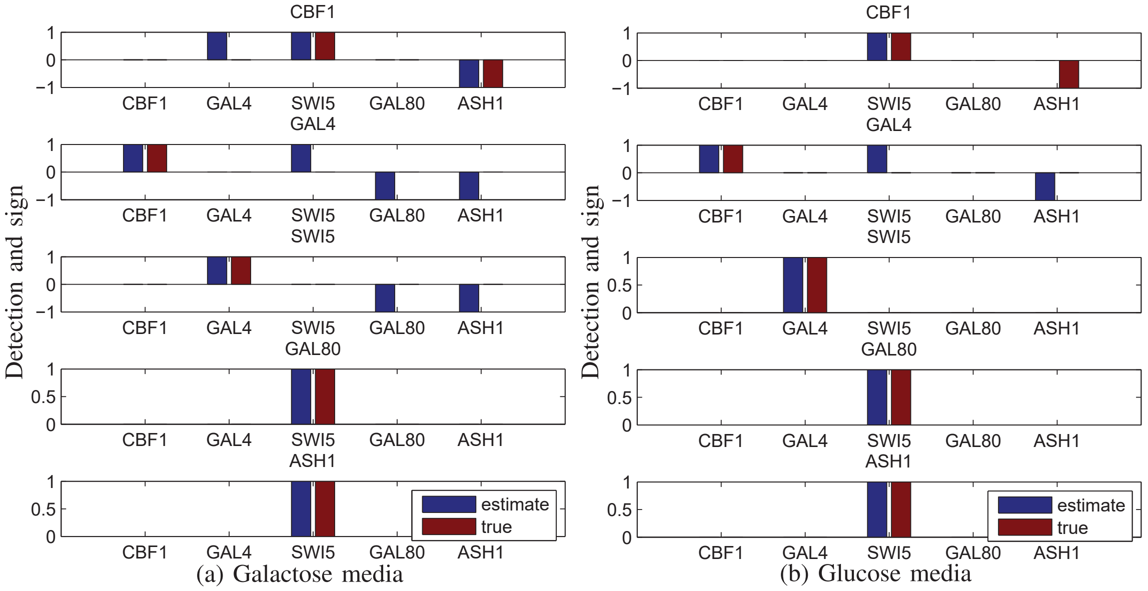
We also implemented the ℓ_1 regularization method. Unfortunately, due to numerical difficulties, the algorithm



(a) Galactose media

(b) Glucose media

Fig. 6 Recovery Results: NIR method using data from Fig. 3



(a) Galactose media

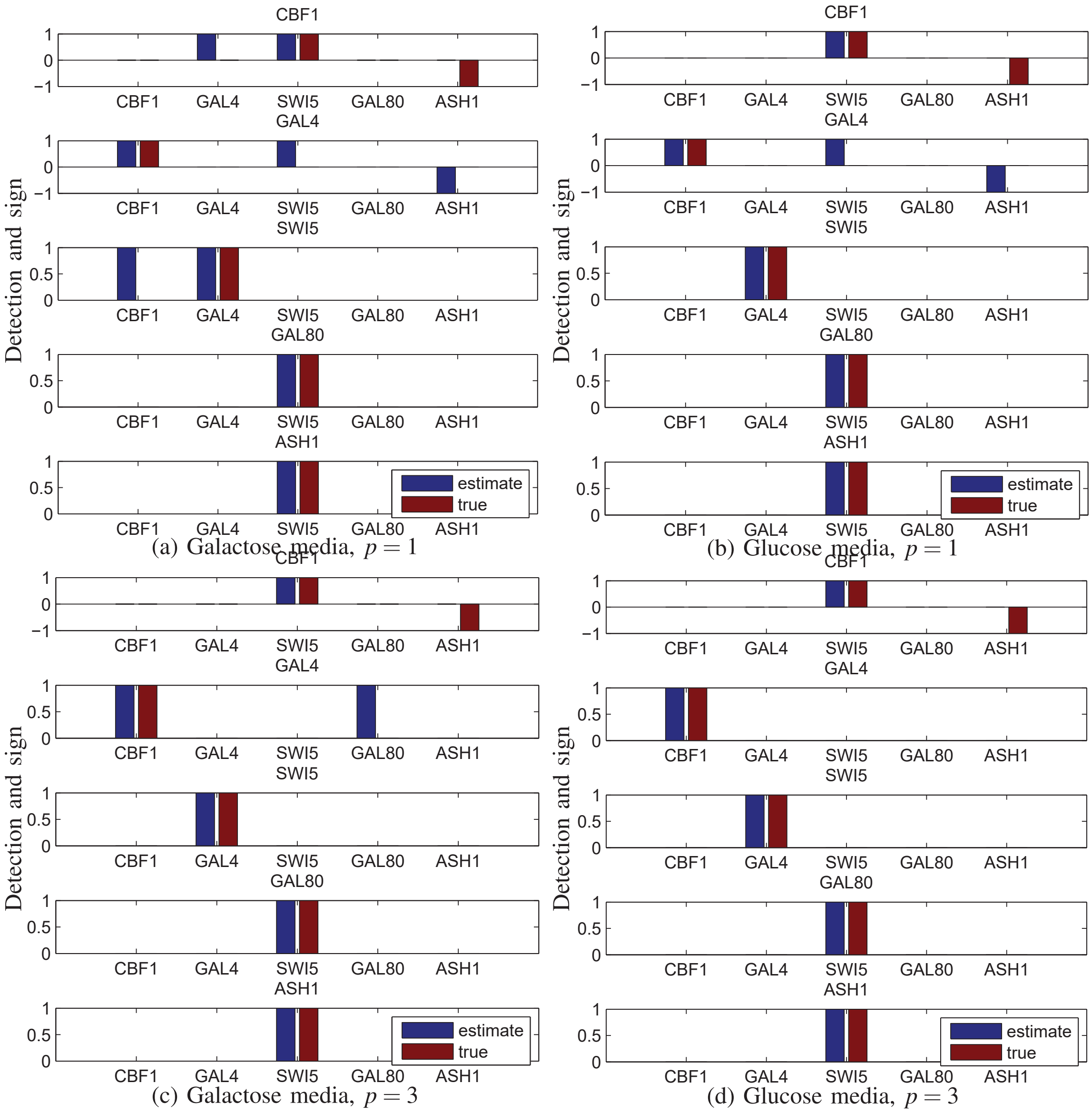
(b) Glucose media

Fig. 7 Recovery Results: ℓ_1 method using data from Fig. 3

TABLE I

Parameters for Synthetic Yeast Network of [1]

Parameter	Value	Parameter	Value
α_1	0	v_1	0.04
α_2	1.49×10^{-4}	v_2	8.82×10^{-4}
α_3	3.0×10^{-3}	v_3	0.0201 (Gal.) .0022 (Gluc.)
α_4	7.4×10^{-4}	v_4	0.0147
α_5	6.1×10^{-5}	v_5	0.0182
k_1	1	d_1	0.0222
k_2	0.0356	d_2	0.0478
k_3	0.0372	d_3	0.4217
k_4	.01 (Gal.) 0.0938 (Gluc.)	d_4	0.010
k_5	1.814	d_5	0.05
k_6	1.814		
h_1,h_2,h_3,h_5,h_6	1	h_4	4
β_1	.2014	γ	0.6 (Gal.) 0.2 (Gluc.)
β_2	.1676	τ	100



(a) Galactose media, $p = 1$

(b) Glucose media, $p = 1$

(c) Galactose media, $p = 3$

(d) Glucose media, $p = 3$

Fig. 8 Recovery Results: Our method using data from Fig. 3

had to be modified so that the errors were measured in Frobenius norm (i.e. some of squared elements,) rather than the sum of absolute values. Specifically, we used the constraint

$$\|XA^T + U\|_F \le \varepsilon.$$

We used $\delta = .001$ and $t = .1$. The results are shown in Fig. 7. Since the estimated A was stable, no additional iterations using stability constraints were taken. In this plot, the connectivity is determined via the size of the weight w_{ij} and the sign of a_{ij}. For each i (each row of A) we find

$$w_{min} = \min_{j \neq i} w_{ij}$$

and select gene j as an input to gene i if $w_{ij} < 10w_{min}$. The gene is activating if A_{ij} has positive sign, and inhibitory if A_{ij} has negative sign.

Next we implemented our method with $p = 1$ and $p = 4$ (the number of saturation functions). The same value for t and δ was selected, and the same method for selecting input genes, except the sign is chosen based on the largest element of $(A_e^T)^{i\downarrow}_{\Omega_j}$. The results are shown in Fig. 8. Note that while the $p = 1$ results (which assumed linear dynamics) is similar to the ℓ_1 results, with $p = 3$ there are a total of 1 false positive and 2 false negatives, which is an improvement over both NIR and the linear ℓ_1 method, especially for the Galactose media.

V. Discussion

More accurate models, including, for example, a detailed description of the galactose system, or those based on different formalisms, can be developed, depending on the biological question to be investigated, and assessed against the same ground truth provided by the IRMA network. In [8], the Huber group LASSO and the group LASSO are applied to the IRMA data in three cases: (1) switch on datasets, (2) switch off datasets and (3) all datasets, i.e., combining switch on and switch off datasets. In the stability selection procedure, the number of bootstrap samples is 30 for all cases and the moving block length is 14 for the second case and 8 for the other cases. To quantify the prediction accuracy, two measures are used: the *area under the receiver operating characteristics* (AUROCs) and *precision-recall curves* (PRCs). Group LASSO does not predict well for the switch on datasets but the predictive performances of these methods are otherwise better than random guesses. The Huber group LASSO outperforms the group LASSO in both AUROCs and AUPRs. All methods for the switch off datasets perform better than for the switch on datasets. The group LASSO for all datasets has better performance than for the switch on datasets but is not as good as for the switch off datasets. The Huber group LASSO for all datasets has the best performance among all cases. This indicates that combing multiple datasets may lead to either the best result or a robust result which is better than the worst case. The network topology with *false positive rate* (FPR) 0.08 of the Huber group LASSO for all datasets is shown in [8] and the corresponding *true positive rate* (TPR) is 0.75 with precision 0.86, in which the red edges represent true positives while black edges are false positives. The results show the effectiveness of our method for the IRMA data. Note that all methods above implicitly assume a linear model. In the approach detailed in the next section, we explicitly account for the fact that the true model has a very characteristic nonlinearity.

VI. Conclusion

In [1], a celebrated framework was established that showed how a semi-quantitative prediction of cell behavior is possible, even with a simplified phenomenological differential equation model. We have improved upon its predictive algorithm by using piecewise affine nonlinear functions as the basis functions. One of the difficulties in obtaining a predictive and quantitative model in biology is the choice of the unknown kinetic parameters, especially for even a mildly complex networks such as the IRMA network (33 parameters) of [1]. A different set of parameters may yield similar results. Ideally, the kinetic parameters should be identified by appropriate experiments, and this is not always possible, particularly if one wants to obtain quantitative values. Here, we were able to measure, semi-quantitatively, the strength of the promoters, and we estimated 16 out of 33 parameters from these data. Remarkably, despite all of the simplifications made, the model showed predictive power, albeit semi-quantitative. All in all, whereas the algorithms of [1] predicted 9 interconnections incorrectly in the 30-node IRMA network, our algorithm predicts only 3 interconnections incorrectly.

References

[1] I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. D. Bernardo, and M. P. Cosma, “A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches,” *Cell*, vol. 137, pp. 172–181, 2009.

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

[3] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke, “Gene regulatory network inference: Data integration in dynamic models - a review,” *BioSystems*, pp. 86–103, 2009.

[4] D. Camacho and J. Collins, “Synthetic biology strikes gold,” *Cell*, vol. 137, no. 1, pp. 24–26, 2009.

[5] 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*, pp. 102–105, 2003.

[6] A. Julius, M. Zavlanos, S. Boyd, and G. Pappas, “Inferring genetic networks and identifying compound mode of action via expression profiling,” *Automatica*, vol. 47, no. 6, pp. 1113–1122, 2011.

[7] ——, “Genetic network identification using convex programming,” *Systems Biology, IET*, vol. 3, no. 3, pp. 155–166, 2009.

[8] L.-Z. Liu, F.-X. Wu, and W.-J. Zhang, “A group LASSO-based method for robustly inferring gene regulatory networks from multiple time-course datasets,” *BMC Systems Biology*, vol. 8, no. Supl 3: S1, pp. 1–12, 2014.