A COMPARISON OF NETWORK RECONSTRUCTION METHODS FOR CHEMICAL REACTION NETWORKS

C. Ward[1], E. Yeung[1], T. Brown[1], B. Durtschi[1], S. Weyerman[1], R. Howes[2], J. Goncalves[3], and S. Warnick[1]

Abstract: Chemical reaction networks model biological interactions that regulate the functional properties of a cell; these networks characterize the chemical pathways that result in a particular phenotype. One goal of systems biology is to understand the structure of these networks given concentration measurements of various species in the system. Previous work has shown that this network reconstruction problem is fundamentally impossible, even for simplified linear models, unless a particular experiment design is followed. Nevertheless, reconstruction algorithms have been developed that attempt to approximate a solution using sparsity or similar heuristics. This work compares, in silico, the results of three of these methods in situations where the necessary experiment design has been followed, and it illustrates the degradation of each method as increasing noise levels are added to the data.

Keywords: Network Reconstruction, Chemical Reaction Networks, Optimization, Linear Programming, Linear Matrix Inequalities, in Silico Comparison.

Introduction: Network Reconstruction

The life processes of all organisms are carried out by chemical reaction networks within and between that organism's cells. These networks consist of complex interactions among proteins, nucleic acids, and other molecules [1], [2], [3]. Often very little is understood about the underlying structure and function of these networks. This contributes to the inability to uncover the causes of many diseases or design a cure for them. Although recent developments in biotechnology, such as DNA microarray, protein-protein interaction measurements, and protein chips, have produced massive amounts of quantitative data, there is still much to discover about how to use this data to understand the underlying networks [4], [5].

One goal of systems biology is to understand the structure of these networks given concentration measurements of various species in the system. This network reconstruction problem is shown to be impossible, even for simplified linear models, unless a particular experiment design is followed [6]. This essential experiment design demands that a particular input output experiment is conducted for every node in the desired network representation of the system. Thus, for example, if the system actually involves reactions between hundreds or thousands of distinct chemical species, we may only measure the concentrations of some limited number of these species. The network structure we are looking for, then, is the causal relationship between the concentrations of these measured species--each of which may involve a pathway through a number of other unmeasured species in the system. Note that since these intermediate species are unmeasured, we may not even be aware of their existence or of their role in the various pathways impacting the measured species

Nevertheless, the network structure we desire is not necessarily the direct interaction between measured species, but it may involve a complex pathway through unmeasured species that we simply indicate as a connection between the relevant measured species. This network would be the complete chemical reaction network if all species in the system are measured, but it simplifies to a proper abstraction of this network commensurate with the number of unmeasured, or "hidden" states in the system.

Reconstructing this "coarse" structural representation of the "complete" network demands that a particular input-output experiment is performed for each measured species, or node, of the final network. Specifically, the experiment for each node needs to perturb the measured species associated with that node without affecting the other measured species, as happens with silencing or inducible overexpression experiments for gene regulatory networks. If such data is available, network reconstruction reduces to performing a sequence of system identification experiments and solving a linear system of equations. If such data is not available, then one can demonstrate multiple structures that fit the data equally well.

Various structure estimation algorithms have been developed without exploiting this understanding of the necessary experiment design for reconstruction. These methods resolve the ambiguity resulting from insufficient and properly-informative data by enforcing a heuristic on the kinds of structures (such as sparse structures) believed to be most likely to occur in nature.

The goal of this study is to compare the performance of three network reconstruction methods, assuming the appropriate experiment design is employed, ensuring that sufficiently

[1] IDeA Labs, Brigham Young University. Please address all correspondence to C. Ward at candice144@gmail.com or S. Warnick at sean.warnick@gmail.com

[2] UCLA

[3] Cambridge University, UK

informative data is available for reconstruction—at least in the noise-free setting. The first two methods employ heuristics such as sparsity to formulate convex optimization problems that select a structural estimate given input-output data from the system. The third method computes the exact network structure, assuming no noise in the system, even though noise may actually be present. We compare these methods in various situations of differing numbers of hidden states, and for differing amounts of noise. Similar analyses of network reconstruction methods have been performed [7], [8], but none have concerned the particular methods discussed in this paper. The next section describes the methodology for the in silico comparison experiment.

Methodology: In Silico Experimentation

The idea behind testing the reconstruction methods considered here is to 1) randomly generate various network models, 2) simulate the data for various "silencing" and wild type experiments from each model, and then 3) compare the performance of three reconstruction algorithms estimating the structure of the original network for each model. Repeating the experiment for various types of networks then facilitates an understanding of how the algorithms perform in different application domains.

Randomized Network and Data Generator

The heart of the in silico experimental process is a routine that generates random networks and their corresponding time series concentration data for a complete set of silencing and wild-type experiments. Motivated by basic reaction kinetics, we represent chemical reaction networks by a system of coupled differential equations. Moreover, we restrict our attention to linear networks here, since it seems that any reconstruction method that will work on real biological systems should at least perform well on simple, linear systems.

With this restriction to linear networks, our model becomes:

$$\frac{dx}{dt} = Ax(t)$$

$$y(t) = [I \quad 0]x(t) + d\eta(t)$$

where $I \in \mathrm{R}^{pxp}$ is the identity matrix, $x(t) \in \mathrm{R}^{nx1}$ is a vector of chemical concentrations for each species in the system, $y(t) \in R^{px1}$ is a vector of chemical concentrations for the p measured species in the system, $\eta(t) \in R^{px1}$ is a random noise vector, and d is a scalar weight on the noise vector. The matrix A is generated to ensure that 1) the system is stable, 2) the hidden states of the system are observable, and 3) the resulting network structure between measured states reflects a desired level of connectivity. The system is then simulated from a random initial condition, perturbed by noise generated at the desired signal to noise ratio, to yield 10 sets of wild-type data for the system. Data for a complete set of silencing experiments is obtained from the system as follows. The state vector $x(t)$ is augmented to include the concentration $z(t)$of a transgene. The system dynamics are then augmented as

$$\frac{d\begin{bmatrix}x(t)\\z(t)\end{bmatrix}}{dt} = \begin{bmatrix}A & e_i\\0 & -\alpha\end{bmatrix}\begin{bmatrix}x(t)\\z(t)\end{bmatrix}$$

where e_i is a vector of zeros everywhere except in location i, and α is a degradation constant modeling the action of the transgene. Resimulating the augmented system then yields the output when species i is silenced, and this process is repeated for all p measured states in the system.

Thus, for each randomly generated network, a set of $p+10$ datasets is provided: ten wild-type data sets in addition to data for p silencing experiments. Each algorithm uses a different subset of this data. The exact reconstruction algorithm uses the p silencing experiments, the LMI-based algorithm uses one wild-type experiment and the linear programming sparsity heuristic uses 10 wild type data sets.

Although each of the algorithms requires different types of data, they all share the property that they take in time-series data and output a network structure. Hence, we are able to compare the accuracy of the algorithms by the number of correctly predicted connections in the adjacency matrix.

Three reconstruction methods are then employed: exact reconstruction, a linear programming sparsity heuristic, and an LMI-based algorithm.

Exact Reconstruction

The exact reconstruction method refers to the results from [6] that demonstrate precisely how to obtain network structure for linear networks with no noise. The methodology leverages knowledge about the perturbation, i.e. silencing, experimentation process to solve for the dynamical structure function of the system. The Boolean structure of this function then reveals the presence or absence of edges in the network.

To solve for the complete dynamical structure function of the system, we employ standard system identification methods, such as those from the MatLab System Identification Toolbox, to identify a transfer function associated with each of the p silencing experiments performed on the system. These transfer functions can then be manipulated to find the dynamical structure function of the system.

A quicker method employs only the steady state values from the datasets to find the value of the transfer function at zero, yielding the Boolean structure of the network without computing the entire dynamical structure function. Although there are variations to this method that consider noisy data, the version employed here is designed to exactly reconstruct a network in the idealized setting i.e. for linear systems with no noise.

A Linear Programming Sparsity Heuristic

Another approach to reconstruction solves a 1-norm regression problem to reward sparse structure estimates [5]. This method is designed for linear and non-linear networks. Network structure is formulated as an adjacency matrix $K \in R^{nxn}$of reaction kinetic coefficients. Treating these coefficients as decision variables, structure is derived by solving a linear program for each of ten sets of wild-type time series data. The final structure estimate is written as a *probability lookup table,* given by

$$\overline{K} = \frac{1}{10}\sum_{i=1}^{10} B(K_i)\,,$$

where the entries of K_i are the estimated reaction coefficients of the network, and $B(\cdot)$ denotes the Boolean structure. Decision variables k_{ij} corresponding to zero entries in $\overline{K}$ are set to be zero and the linear program is solved again for the same ten data sets to obtain a richer sparsity structure. In addition, entries of the resulting matrix of fractional occurrences which are less than a threshold are set to zero.

This linear programming technique was solved using the cvx toolbox [10], [13].

Alternative techniques are described, such as a robust formulation for uncertainties in the data and a formulation to account for species which could not be measured due to technical reasons. These techniques will be the subject of future work.

LMI-Based Algorithm for the Reconstruction of Biological Networks

This method reconstructs an adjacency matrix *A* describing the network structure among *n* chemical species satisfying a discrete-time system of *n* difference equations [11], namely

$$x(k+1) = Ax(k)\,, \quad A \in R^{nxn}$$

where the concentrations of each chemical species is stored in the vector

$$x(k) = [x_1(k) \;\; \ldots \; x_n(k)]^T$$

Writing the time series data from one wild type experiment, as an *h* dimensional vector,

$$\Theta := \begin{bmatrix} x(h) \\ \vdots \\ x(1) \end{bmatrix} = A\Omega,$$

where

$$\Omega := \begin{bmatrix} x(h-1) \\ \vdots \\ x(0) \end{bmatrix}.$$

The object is to reconstruct the matrix *A* from experimental values ($x(h) \;\; \ldots \;\; x(0)$). Mathematically this translates into minimizing the norm of $\Theta - A\Omega$ or, equivalently,

$$\min_{A} \varepsilon \quad s.t.$$

$$(\Theta - A\Omega)^T (\Theta - A\Omega) < \varepsilon I$$

Through the mechanics of Schur complements this nonlinear constraint is transformed to the following equivalent, linear constraint

$$\begin{bmatrix} -\varepsilon I & (\Theta - A\Omega)^T \\ (\Theta - A\Omega) & -I \end{bmatrix} < 0$$

This results in a generalized eigenvalue problem which can be solved through the use of such tools as in the Matlab LMI Toolbox. The method proposed by these authors also allows for *a priori* knowledge of a biological network to be included in the optimization problem. Such information can be included in the form

$$u_i{}^T A u_j + u_j{}^T A^T u_i > 0 \quad (resp. < 0),$$

where $u_k := [0 \;\; \ldots \;\; 1 \;\; \ldots \;\; 0]$ is a column vector of zeros, with a 1 at the *k*-th position. Also, a zero coefficient can be included by simply removing the corresponding variable from the problem and substituting it with zero. Furthermore, this method not only aims to minimize the interpolation error, but also attempts to minimize the number of nonzero entries in the *A* matrix. This is based on the assumption that biological networks are generally sparse.

The implementation of this method is then as follows:

1) Solve problem to identify an *A* matrix from the experimental data.
2) Normalize the *A* matrix by dividing each of its entries by the magnitude of the corresponding row and column.
3) Analyze the normalized matrix. Entries smaller than a determined threshold are nullified.
4) The LMI method is set up based on the resulting matrix and a new solution is computed.
5) Time series data of the new solution is compared to that of the experimental data. If the error between the two is too big, the method stops. Otherwise, another iteration begins at 2).

Numerical Study

We simulated 900 networks with 15 chemical species, divided into 9 network categories. Each network category was specified by 2 parameters: *p*, the total number of measured species and *SNR*, the signal-to-noise ratio. Using the time-series data described above, each method

Table 1

p	SNR	LP		LMI		ER	
		SS	SP	SS	SP	SS	SP
9	*No noise*	0.360	0.828	0.889	0.499	0.709	0.966
	1	0.250	0.902	0.889	0.510	0.520	0.737
	1/100	0.445	0.818	0.901	0.517	0.479	0.754
12	*No noise*	0.298	0.762	0.894	0.274	0.740	0.967
	1	0.206	0.851	0.899	0.278	0.492	0.598
	1/100	0.487	0.736	0.878	0.282	0.499	0.606
15	*No noise*	0.313	0.789	0.903	0.185	0.985	0.977
	1	0.177	0.866	0.882	0.193	0.503	0.554
	1/100	0.391	0.706	0.897	0.181	0.502	0.570

reconstructed 900 networks. Based on the results, sensitivity (SS) and specificity (SP) scores were calculated by

$$SS = \frac{TP}{TP + FN}, \qquad SP = \frac{TN}{TN + FP}$$

The scores are shown in Table 1. Sensitivity reports the percentage of edges correctly predicted and specificity reports the percentage of zeros correctly predicted [11].

Our object was to see how performance degraded as the noise level in the data increased. As expected, the exact reconstruction method performed well with no noise. This method identified zeros better than edges, partially because of implementation issues. Since we encountered difficulties with symbolic matrix inversion, we implemented the steady-state approach to just recover Boolean structure. The performance dropped significantly with noisy data. Additional noise did not lower the performance.

The sparsity heuristic had a high specificity score, reflecting its ability to produce sparse solutions. With higher levels of noise, this method faithfully predicted the absence of edges and surprisingly predicted the presence of edges with increasing accuracy. Research examining how this algorithm performs for different levels of sparsity will be explored in the future.

The LMI-based algorithm tended to have high sensitivity scores, indicating highly connected solutions. The algorithm fits a matrix to the time series data, thus, highly connected networks are favored since they have more degrees of freedom to fit the data. The transition from noise-free data to noisy data was characterized by a significant drop in the algorithm's ability to determine the absence of edges.

With each method, performance dropped as the data became noisy. In the case of the sparsity heuristic and LMI method, the drop was only noticeable in the sensitivity and specificity categories respectively. The exact reconstruction technique performed noticeably worse in both SS and SP categories with the addition of noise.

Conclusion

This paper analyzed the quality of three network reconstruction methods under varying levels of noise and hidden states. We have found that with the addition of noise adversely affected the performance of each algorithm; with the most drastic effect on the exact reconstruction method. The LMI and sparsity method tended to have either dense or sparse solutions, yielding strong sensitivity and specificity performance respectively.

Future work will involve implementing the robust formulation of the sparsity heuristic, applications of model reduction techniques to identify network structure in noisy environments, and assessment of these methods for non-linear systems, varying levels of sparsity, and large networks.

References

[1] T. Ito et al., "Toward a Protein-Protein Interaction Map of the Budding Yeast: A Comprehensive System to Examine Two-Hybrid Interactions in All Possible Combinations between the Yeast Proteins," *Proc. Nat'l Academy of Sciences USA,* vol. 97, pp.1143-1147, 2000.

[2] V. Spirin and L. A. Mirny, "Protein Complexes and Functional Modules in Molecular Networks," *Proc. Nat'l Academy of Sciences USA,* vol. 100, pp. 12123-12128, 2003.

[3] W. Li et al., "Dynamical Systems for Discovering Protein Complexes and Functional Modules from Biological Networks," *IEEE/ACM Trans. on Computational Biology and Bioinformatics,* vol. 4, no. 2, pp. 233-250, 2007.

[4] K.-H. Cho et al., "Reverse engineering of gene regulatory networks," *IET Systems Biology,* vol. 1, pp.149-163, 2007.

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

[6] J. Gonçalves, S. Warnick, "Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks," in *IEEE Transactions of Automatic Control*, 2007.

[7] D. Camacho, P. Licona, P. Mendes, R. Laubenbacher, "Comparison of Reverse-Engineering Methods Using an *in Silico* Network," *Annals of the New York Academy of Sciences,* 2007.

[8] P. Mendes, W. Sha, K. Ye, "Artificial gene networks for objective comparison of analysis algorithms," *Bioinformatics*, 2003.

[9] R. Howes, L. Eccleston, J. Gonçalves, G. Stan, S. Warnick, "Dynamical structure analysis of sparsity and minimality heuristics for reconstruction of biochemical networks," in *Proceedings of 47th IEEE Conference on Decision and Control 2008.*

[10] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming(web page and software) http://stanford.edu/~boyd/cvx, February 2009.

[11] 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.

[12] J. Goncalves, R. Howes, and S. Warnick, "Dynamical Structure Functions for the Reverse Engineering of LTI Networks," *Proc. Conference on Decision and Control*, pp. 1516-1522, New Orleans, LA, 2007.

[13] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs, *Recent Advances in Learning and Control(a tribute to M. Vidyasagar),* V. Blondel, S. Boyd, and H. Kimura, editors, pages 95-110, *Lecture Notes in Control and Information Sciences,*Springer, 2008, http://stanford.edu/~boyd/graph_dep.html.