JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 17, Number 2, 2010

Pp. 1–8
DOI: 10.1089/cmb.2009.0052

Research Article

Reducing the Computational Complexity of Information Theoretic Approaches for Reconstructing Gene Regulatory Networks

PENG QIU, ANDREW J. GENTLES, and SYLVIA K. PLEVRITIS

ABSTRACT

Information theoretic approaches are increasingly being used for reconstructing regulatory networks from microarray data. These approaches start by computing the pairwise mutual information (MI) between all gene pairs. The resulting MI matrix is then manipulated to identify regulatory relationships. A barrier to these approaches is the time-consuming step of computing the MI matrix. We present a method to reduce this computation time. We apply spectral analysis to re-order the genes, so that genes that share regulatory relationships are more likely to be placed close to each other. Then, using a "sliding window" approach with appropriate window size and step size, we compute the MI for the genes within the sliding window, and the remainder is assumed to be zero. Using both simulated data and microarray data, we demonstrate that our method does not incur performance loss in regions of high-precision and low-recall, while the computational time is significantly lowered. The proposed method can be used with any method that relies on the mutual information to reconstruct networks.

Key words: algorithms, computational molecular biology, machine learning.

1. INTRODUCTION

Gene expression microarray technology measures the expression levels of thousands of genes simultaneously and provides data for reconstructing large-scale gene regulatory networks. Recently, information theoretic approaches have been used increasingly for this purpose. Examples include the relevance network (RelNet) (Butte and Kohane, 2000; Butte et al., 2000) ARACNE (Margolin et al., 2006a), and the maximum relevance minimum redundance network (MRNet) (Meyer et al., 2007). These approaches and others start by computing the pairwise mutual information (MI) between all possible pairs of genes, resulting in a MI matrix. The MI matrix is then manipulated to identify regulatory relationships. For example, in relevance networks, an edge exists between a pair of genes if their MI exceeds a given specific threshold. ARACNE improves on the relevance networks by applying the data processing inequality (Cover and Thomas, 2006) (DPI) to each connected gene triple, removing potentially false positive edges. In MRNet, the maximum relevance minimum redundance (MRMR) criterion is applied, where the maximum relevance

Department of Radiology, Stanford University, Stanford, California.

criterion assigns edges to gene pairs that share large MI, while the minimum redundance criterion controls false positives (Meyer et al., 2007; Ding and Peng, 2005). The MRMR criterion is essentially a pairwise approximation of the conditional MI. The conditional MI is explicitly used for inference in Zhao et al. (2008). If two genes that share large MI but are conditionally independent give a third gene, there is no edge between them. These approaches have been successfully applied to simulated data and real microarray data on B-Cell, melanoma, and have identified interesting regulatory targets and pathways.

The most time-consuming step in all of the above approaches (Butte and Kohane, 2000; Margolin et al., 2006a; Meyer et al., 2007) is computing the entire MI matrix, which requires all possible pairs of genes to be examined. The computational complexity of Zhao et al., (2008) is higher because the conditional MI has to be computed for each gene triple. To reduce the computational complexity, we only compute the MI for gene pairs with expected significant values. We identify these gene pairs by applying spectral analysis (Chung, 1997) to re-order the genes, so that genes that share regulatory relationships are more likely to be placed close to each other. To determine the new gene ordering, a Laplacian matrix is derived from the correlation matrix of the gene expression data, assuming the correlation matrix provides an adequate approximation to the adjacency matrix for our purpose. Motivated by the spectral decomposition in Rapaport et al. (2007), we then compute the Fiedler vector, which is the eigenvector associated with the second smallest eigenvalue of the Laplacian matrix. Since the Fiedler vector is smooth with respect to the connectivity described by the Laplacian matrix (Rapaport et al., 2007), we sort the elements of the Fiedler vector to obtain the desired gene ordering.

After re-ordering the genes, we apply a sliding window approach with respect to the new gene ordering and compute the MI among the genes within the sliding window. Since we only compute the MI among genes within the sliding window, only part of the MI matrix is computed and the remainder is assumed to be zero. The resulting MI matrix can then be used with a variety of information theoretic approaches for reconstructing the regulatory network, including, for example, RelNet, ARACNE, and MRNet. Depending on the window size, we demonstrate that the computational complexity of computing the MI matrix can be significantly reduced, with minor loss in the accuracy of the reconstructed regulatory network.

2. METHODS

2.1. Re-ordering the genes

A gene regulatory network reconstructed by the information theoretic approaches (Butte and Kohane, 2000; Margolin et al., 2006a; Meyer et al., 2007) is undirected and can be represented by an adjacency matrix. The adjacency matrix is usually sparse, and a value of 1 at the (i,j) element represents an edge connecting gene i and gene j. Since the adjacency matrix is sparse, its rows and columns can be reshuffled (resulting in a re-ordering of the genes), so that most 1's are concentrated along the diagonal of the reshuffled adjacency matrix.

We apply spectral analysis to re-order the genes to obtain an adjacency matrix with the 1's concentrated along the diagonal. For example, consider Figure 1a, which shows the adjacency matrix of a simulated scale-free network, containing 100 genes and 135 edges. For the moment, assume that we know the adjacency matrix A. We define the Laplacian matrix as $L = D - A$, where D is a diagonal matrix whose diagonal elements are the degrees of the genes. In spectral graph theory (Chung, 1997), the Fiedler vector, $\mathbf{v} = [v_1, v_2, \ldots, v_n]'$, is defined as the eigenvector associated with the second smallest eigenvalue of the Laplacian matrix (the smallest eigenvalue is trivial). As presented in the spectral decomposition (Rapaport et al., 2007), the Fiedler vector is smooth with respect to the network connectivity described by the Laplacian matrix. By "smooth with respect to the network," we mean that if two nodes i and j are connected by an edge, the difference between their corresponding entries in the Fiedler vector, $|v_i - v_j|$, is small. The mathematics behind the above argument is as follows. Since the Fiedler vector is the first non-trivial eigenvector associated with the smallest non-trivial eigenvalue of L, the Fiedler vector solves the following minimization problem,

$$\begin{aligned} \min \quad & \mathbf{v}' L \mathbf{v} \\ s.t. \quad & |\mathbf{v}| = 1 \end{aligned} \tag{1}$$

where the objective function can be written as follows,

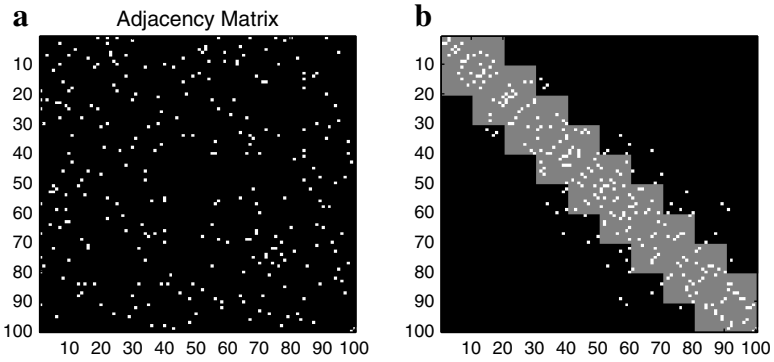

FIG. 1. (**a**) The adjacency matrix of a simulated scale-free network, consisting of 100 genes and 135 edges. (**b**) The reshuffled adjacency matrix using spectral analysis. With a sliding window of size 20 and step size equal to half of the window size, the sliding window approach covers 28% of the adjacency matrix (gray area), which contains 85% of the edges.

$$\begin{aligned}\mathbf{v}'L\mathbf{v} &= \mathbf{v}'D\mathbf{v} - \mathbf{v}'A\mathbf{v} \\ &= \sum_{i\in\{1,\ldots,n\}} d_i v_i^2 - \sum_{i,j\in\{1,\ldots,n\}} v_i A_{ij} v_j \\ &= \sum_{\{i,j:i>j,A_{ij}=1\}} (v_i^2 + v_j^2) - \sum_{\{i,j:i>j,A_{ij}=1\}} 2v_i v_j \\ &= \sum_{\{i,j:i>j,A_{ij}=1\}} (v_i - v_j)^2 \end{aligned} \tag{2}$$

Therefore, the Fielder vector is also minimizes (2), which means if two nodes i and j are connected, the difference between their corresponding entries in the Fiedler vector, $|v_i - v_j|$, tends to be small.

We compute the Fiedler vector of the Laplacian matrix L and sort its elements in either ascending or descending order. In the gene ordering obtained from the Fiedler vector, connected genes are more likely to be placed close to each other. Figure 1b shows the reshuffled adjacency matrix, with most edges (1's) now close to the diagonal. In this particular example, if we apply a window of size 20 and slide the window along the new gene ordering with a step size of half of the window size, the sliding window approach covers 28% of the adjacency matrix (the gray area), but reconstructs 85% of the edges. Although we miss 15% of the edges by applying the sliding window, to ensure high-precision and low-recall, most of these missing edges would not be identified as significant even when the entire MI matrix is computed. Simulation results in Section 3 show that applying the sliding window approach does not incur performance loss in operating regions of high-precision and low-recall.

In practice, the adjacency matrix is an unknown, so we need to obtain the gene re-ordering from the expression data. We proceed by assuming that the correlation matrix provides an approximation of the adjacency matrix that is adequate for our purpose, because the correlation matrix reflects the linear relationship between gene pairs and is fast to compute. We normalize each gene's expression data to zero mean and unit variance, and compute the correlation matrix by multiplying the expression data and its transpose. The power adjacency function (Zhang and Horvath, 2005) is applied to each entry of the correlation matrix ($y = |x|^K$). The resulting matrix, denoted by W, is an approximation to the adjacency matrix. We define the Laplacian as $L = D - W$, where D is a diagonal matrix, chosen such that the column sums of L are zeros. Eigen-decomposition of the Laplacian is performed to find the Fiedler vector, which is sorted to re-order the genes.

The computational complexity of obtaining the gene ordering is negligible compared to the computation of the MI matrix. For example, for a B-Cell gene expression dataset (Margolin et al., 2006a), which contains 336 samples and 9563 genes per sample, computing the entire MI matrix takes about 142 hours, using the Java package developed in Margolin et al. (2006b), whereas obtaining the gene ordering using the above procedure takes less than 10 minutes in Matlab.

2.2. Reducing computation time by the sliding window approach

After re-ordering the genes, connected genes are more likely to be placed close to each other. We then apply a sliding window along the new ordering, and compute MI within the sliding window. This concept

can be visualized in Figure 1b. If we reshuffled the genes according to the ordering, the gene pairs within the sliding window correspond to the gray area, along the diagonal line of the reshuffled MI matrix. The blocks are determined by the window size and the step size of the sliding window. With an appropriate window size and step size, the computational complexity can be significantly reduced with minimal performance loss.

The reduction in computational complexity can be quantified as follows. Assume that the total number of genes is n, the size of the sliding window is w, and the step size is chosen to be half of the window size. To compute the entire MI matrix, n^2 pairs of genes need to be examined. The number of gene pairs covered by the sliding window is $\frac{w^2}{4} + \frac{3w^2}{4}\frac{(2n-1)}{w}$. The ratio is approximately $\frac{3w}{2n}$. In the example in Figure 1, there are $n = 100$ genes, and the window size $w = 20$. Using the sliding window, the computational complexity is reduced to 30% of that needed to obtain the entire MI matrix.

The reduction in computational complexity is the result of computing only the diagonal part of the reshuffled MI matrix. Because the remaining entries of the MI matrix are set to be zeros, there is potential loss of reconstruction accuracy. In the following section, we examine a simulated dataset and a B-Cell gene expression dataset to evaluate the performance loss.

3. RESULTS

3.1. Simulated data

We use the data generator in Rogers and Girolami (2005) to simulate artificial gene regulatory networks and expression data. The generator produces topologically scale-free networks (Barabási and Bonabeau, 2003), where the number of regulatory connections for each gene is generated according to an approximate power-law distribution. A symmetric adjacency matrix is derived from the simulated topology, serving as the ground truth in the simulation. The expression data is drawn from the steady state of the system, which is evaluated by integrating a system of differential equations. The generator is able to simulate gene knock-out experiments by holding the expression of the knocked-out gene at zero during the integration (Rogers and Girolami, 2005). In our simulation, we use the data generator to produce gene regulatory networks and simulate the expression data of knock-out experiments for each gene. The number of genes n is chosen to be either 100 or 400.

Similar to Margolin et al. (2006a) and Meyer et al. (2007), we use precision and recall as performance metrics. Denote the true positives N_{TP}, the false positives N_{FP}, and the false negatives N_{FN}. The precision is defined as $p = N_{TP} / (N_{TP} + N_{FP})$, which is the fraction of the true edges among the edges identified by the algorithm. The recall is defined as $r = N_{TP} / (N_{TP} + N_{FN})$, which is the ratio between the correctly identified true edges over the total number of true edges. The precision-recall curve is generated by adjusting the MI threshold in the regulatory network reconstruction algorithms.

Figure 2 shows an example of a simulated dataset containing 100 genes and 135 edges. The expression data contains 100 samples, corresponding to the gene knock-out experiment for each gene. Figure 2a shows the simulated adjacency matrix, which is the same as Figure 1a. We compute the correlation matrix of the simulated expression data, and obtain the gene ordering using spectral analysis as described in Section 2. Figure 2b shows the reshuffled adjacency matrix according to the ordering, which is inferior compared to that in Figure 1b. This is because the ordering in Figure 1b is based on the knowledge of the true adjacency matrix, while the ordering in Figure 2b is based on the correlation matrix of the expression data.

In Figure 2c, the performance of RelNet, ARACNE, and MRNet are compared using precision-recall curves. Both ARACNE and MRNet show superior performance to RelNet, which is consistent with the results in Margolin et al. (2006a) and Meyer et al. (2007). In Figure 2d–f, the three methods—RelNet, ARACNE, and MRNet—are compared with their windowed versions—win-RelNet, win-ARACNE, and win-MRNet, respectively. We apply a sliding window along the gene ordering in Figure 2b, with a window size of 10 and a step size of 5. Such choice of the sliding window reduces the computational complexity to 15% of that when the entire MI matrix is computed. We observe that in the high-precision low-recall regime, applying the sliding window does not incur performance loss. In some cases, applying the sliding window yields slightly better performance. In the low-precision regime, the windowed version has lower recall. However, in practice, this regime is of little value, because it is not possible to distinguish biologically meaningful edges from false positives.

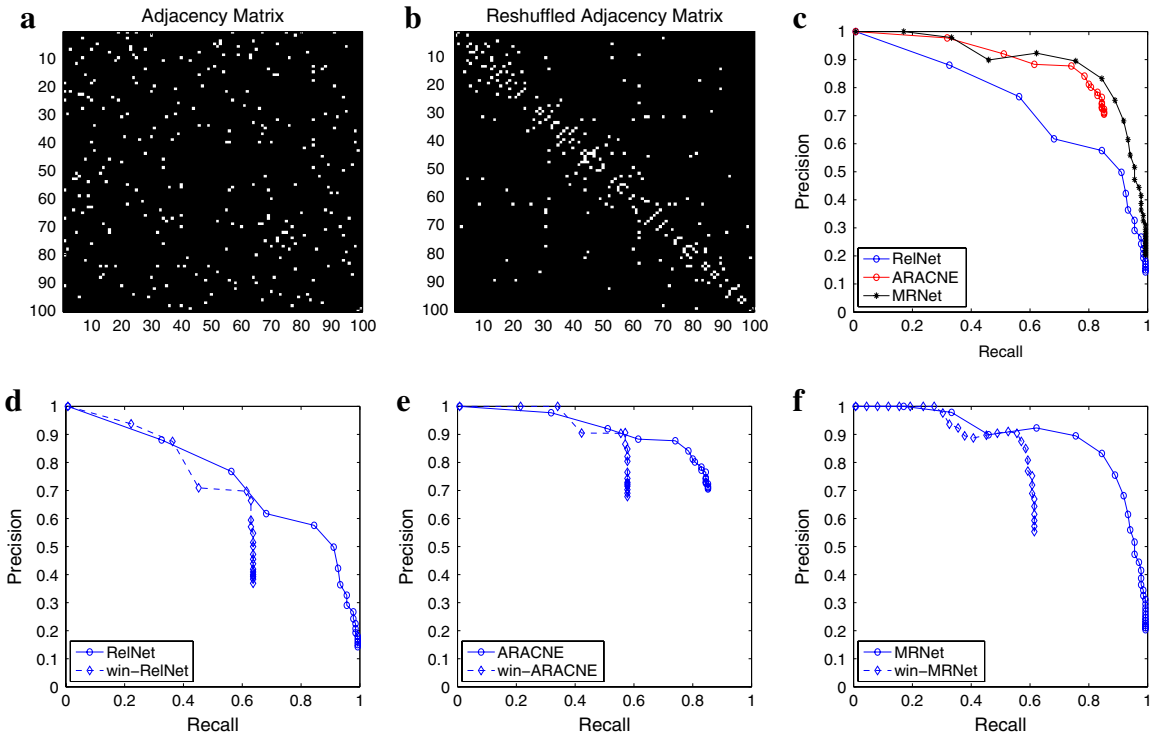

FIG. 2. Performance comparison of RelNet, ARACNE, MRNet, and their windowed version, using a simulated dataset (G100Data01). The number of genes is 100. The sliding window has window size 10 and step size 5. (**a**) Adjacency matrix. (**b**) Reshuffled adjacency matrix. (**c**) Comparison of RelNet, ARACNE, MRNet. (**d**) RelNet versus its windowed version. (**e**) ARACNE versus its windowed version. (**f**) MRNet versus its windowed version.

To further demonstrate that applying the sliding window approach does not incur performance loss in the high-precision low-recall regime, we summarize the results from 10 simulated datasets in Table 1. We integrate the precision-recall curve from 0% to 30% recall, and use the area under curve as the metric of accuracy in the high-precision low-recall regime. We normalize the metric by its maximum possible value 0.3. The first 5 simulated datasets contain $n = 100$ genes and an average of 150 edges. The other 5 datasets contain $n = 400$ genes and around 550 edges. In all cases, the size of the sliding window is chosen to be $w = n/10$, and the step size is half of the window size. Therefore, applying the sliding window reduces the computational complexity to 15% of that when the entire MI matrix is computed. From Table 1, we observe

TABLE 1. PERFORMANCE OF RELNET, ARACNE, MRNET, AND THEIR WINDOWED VERSIONS, WHEN TARGET IS A RECALL OF 30% OR LESS, BASED ON SIMULATED DATA

Simulated dataset	*RelNet*	*win-RelNet*	*ARACNE*	*win-ARACNE*	*MRNet*	*win-MRNet*
G100Data01	0.9462	0.9569	0.9896	1	0.9962	0.999
G100Data02	0.9766	0.9860	0.9841	1	0.9924	1
G100Data03	0.9687	0.9774	1	1	1	1
G100Data04	0.9161	0.9402	0.9633	0.9652	0.9515	0.9313
G100Data05	0.9060	0.9052	0.9586	0.9349	0.9274	0.8925
Average	0.9427	0.9531	0.9791	0.98	0.9735	0.9646
G400Data01	0.8301	0.8267	0.8771	0.8785	0.9234	0.9240
G400Data02	0.8530	0.8336	0.8965	0.8855	0.9304	0.9129
G400Data03	0.846	0.864	0.8912	0.8932	0.9428	0.9345
G400Data04	0.8317	0.8352	0.8783	0.8804	0.9181	0.9147
G400Data05	0.8486	0.8532	0.9007	0.8847	0.9447	0.8994
Average	0.8419	0.8425	0.8888	0.8845	0.9319	0.9171

that the windowed version of RelNet, ARACNE, and MRNet have similar performance compared with their original implementation.

3.2. B-Cell data

We tested the gene ordering and sliding window approach on a publicly available B-Cell gene expression dataset. The dataset contains 336 samples and 9563 genes per sample (Margolin et al., 2006a). Since the true B-Cell gene regulatory network is unknown, precision and recall cannot be used as metric to evaluate the reconstruction performance. Therefore, we only compare the reconstructed networks by the sliding windowed approach to the original implementations of ARACNE.

The gene ordering is obtained as described in Section 2. The window size is chosen to be 1000, and the step size is half of the window size. In Table 2, we show the edges and hub nodes (genes with degree higher than 3) identified by the original implementation of ARACNE and our windowed approach. It can be observed that, when the MI threshold is high, meaning when the algorithm operates at high-precision low-recall regime, there is little difference between the results from ARACNE and our sliding window approach. On average, win-ARACNE recovers 94% of the edges that ARACNE identifies, and at the same time generates $<1\%$ new edges that ARACNE does not identify. win-ARACNE recovers 96% of the hub genes that ARACNE identifies, and does not generate new hub genes that ARACNE does not identify. The computation time for obtaining the gene ordering is less than 10 minutes, which is negligible compared to the 142 hours needed to compute the entire MI matrix. Using the sliding window approach, we reduce the computation time by 84%, that is from 142 to 23 hours.

In order to examine the effect of the window size on the reconstructed regulatory network, we fix the MI threshold to be 0.50, and vary window size from 400 to 1600. For each choice of the window size, we compute what percentage of the edges and hub genes identified by ARACNE are also identified by win-ARACNE. As the window size increases, the regulatory network constructed by win-ARACNE first quickly approaches the network constructed by ARACNE, and then saturates. When the window size increases to 9563, the total number of genes, win-ARACNE and ARACNE should produce identical results. In Figure 3, we also see that the computational complexity is $\frac{3w}{2n}$, linear in the window size w. In summary, we can see that, if we are willing to sacrifice a certain amount of the edges identified by ARACNE, we can apply a sliding window along the proposed gene ordering, and significantly reduce the computational complexity.

4. CONCLUSION

When information theoretic approaches are applied to reconstruct large scale gene regulatory networks, the mutual information between all possible gene pairs needs to be computed, which is a time-consuming

Table 2. Comparison of ARACNE and win-ARACNE on a B-Cell Dataset, which Contains 9563 Genes

	MI threshold				
	0.75	*0.65*	*0.60*	*0.55*	*0.50*
Edges					
ARACNE	1188	3182	6966	13442	23988
win-ARACNE	1134	3058	6666	12630	21638
Overlaps	1134	3054	6656	12596	21328
Hub genes					
ARACNE	61	200	343	558	932
win-ARACNE	61	194	331	524	855
Overlaps	61	194	331	524	855

Window size is chosen to be 1000 and the step size is half of window size. On average, win-ARACNE recovers 94% of the edges that ARACNE identifies, and generates $<1\%$ new edges that ARACNE does not identify. win-ARACNE recovers 96% of the hub genes that ARACNE identifies, and does not generate new hub genes that ARACNE does not identify. Computation time is reduced to 16% of the original implementation.

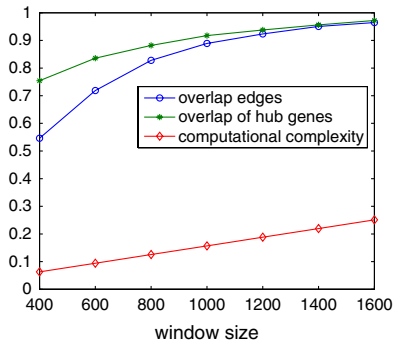

FIG. 3. Comparison of ARACNE and win-ARACNE, in terms of the percentage of overlapping edges and overlapping high degree nodes, and the computational complexity as a function of window size.

task. In this work, we present a method to reduce this computation time. We apply spectral analysis to reorder the genes, so that genes that share regulatory relationships are more likely to be placed close to each other. Using a sliding window with appropriate window size and step size, we can apply the information theoretic approaches to examine the genes subset by subset, along the new gene ordering. Through analysis of simulation and real gene expression data, we demonstrated that, in the high-precision low-recall regime, applying the sliding window approach yields similar performance compared with the case where the entire mutual information matrix is computed, while the computational complexity is significantly reduced.

ACKNOWLEDGMENTS

We gratefully acknowledge funding from the NCI Integrative Cancer Biology Program (ICBP; grant U56 CA112973).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Barabási, A.L., and Bonabeau, E. 2003. Scale-free networks. *Sci. Am.* 288, 60–69.

Butte, A.J., and Kohane, L.S. 2000. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Paci. Symp. Biocomput.* 4, 418–429.

Butte, A.J., Tamayo, P., Slonim, D., et al. 2000. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc. Natl. Acad. Sci. USA* 97, 12182–12186.

Chung, F.R.K. 1997. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92).* American Mathematical Society, New York.

Cover, T., and Thomas, J. 2006. *Elements of Information Theory*, 2nd ed. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, New York.

Ding, C., and Peng, H. 2005. Minimum redundancy feature selection from microarray gene expression data. *J Bioinform. Comput. Biol.* 3, 185–205.

Margolin, A.A., Nemenman, I., Basso, K., et al. 2006a. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinform.* 7.

Margolin, A.A., Wang, K., Lim, W.K., et al. 2006b. Reverse engineering cellular networks. *Nat. Protoc.* 1, 662–671.

Meyer, P.E., Kontos, K., Lafitte, F., et al. 2007. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J. Bioinform. Syst. Biol.* 2007, 9.

Rapaport, F., Zinovyev, A., Dutreix, M., et al. 2007. Classification of microarray data using gene networks. *BMC Bioinform.* 8, 35+.

Rogers, S., and Girolami, M., 2005. A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics* 21, 3131–3137.

Zhang, B., and Horvath, S. 2005. A general framework for weighted gene co-expression network analysis. *Statist. Appl. Genet. Mol. Biol.* 4.

Zhao, W., Serpedin, E., and Dougherty, E.R. 2008. Inferring connectivity of genetic regulatory networks using information-theoretic criteria. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 5, 262–274.

Address correspondence to:
Dr. Peng Qiu
1201 Welch Road, Lucas Center
Department of Radiology
Stanford University
Stanford, CA 94305

E-mail: qiupeng@stanford.edu