

Inferring Genetic Networks from Microarray Data

Shawn Martin, George Davidson,
Elebeoba May & Jean-Loup Faulon
Sandia National Laboratories
Computational Biology
P.O. Box 5800-0310
Albuquerque, NM, 87110, USA
smartin@sandia.gov

Margaret Werner-Washburne
University of New Mexico
Department of Biology
Albuquerque, NM, 87131, USA

Abstract

In theory, it should be possible to infer realistic genetic networks from time series microarray data. In practice, however, network discovery has proved problematic. The three major challenges are 1) inferring the network; 2) estimating the stability of the inferred network; and 3) making the network visually accessible to the user. Here we describe a method, tested on publicly available time series microarray data, which addresses these concerns.

1. Introduction

The inference of genetic networks from genome-wide experimental data is an important biological problem which has received much attention. Approaches to this problem have typically included application of clustering algorithms [6]; the use of Boolean networks [12, 1, 10]; the use of Bayesian networks [8, 11]; and the use of continuous models [21, 14, 19]. Overviews of the problem and general approaches to network inference can be found in [4, 3].

Our approach to network inference is similar to earlier methods in that we use both clustering and Boolean network inference. However, we have attempted to extend the process to better serve the end-user, the biologist. In particular, we have incorporated a system to assess the reliability of our network, and we have developed tools which allow interactive visualization of the proposed network.

2. Network Inference

The first step in our inference algorithm involves clustering the time series microarray data. The clustering algorithm uses force directed graph layout, and produces a two-dimensional representation of the genes from the microar-

ray [2, 13]. In this representation, genes with similar expression profiles are placed near each other, and genes with different expression profiles are placed farther apart. We then partition this representation using the well-known k -means algorithm to provide k groups of co-regulated genes. Altogether, this process not only simplifies the task of network inference (by reducing the problem size), but also results in a network of gene groups, instead of actual genes. These gene groups, which we call *meta-genes*, make the biological analysis and interpretation of the inferred network tractable. Figure (1) illustrates the process of obtaining the gene groups.

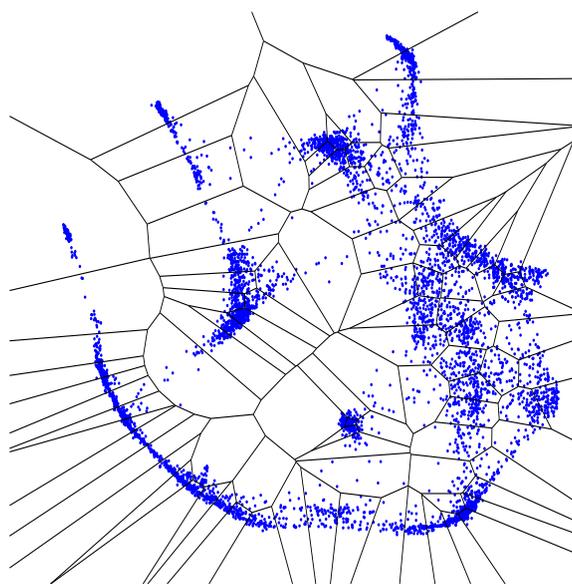


Figure 1. Gene map partitioned by k -means for yeast time series microarray data in [18].

Since our network inference algorithm is Boolean, we must first discretize our the expression levels of our meta-genes. This discretization is accomplished in two steps. First, Support Vector Regression [17] is used to obtain a single continuous curve representing each meta-gene. Next, and on/off expression profile is obtained by thesholding the resulting continuous curve, as shown in Figure (2).

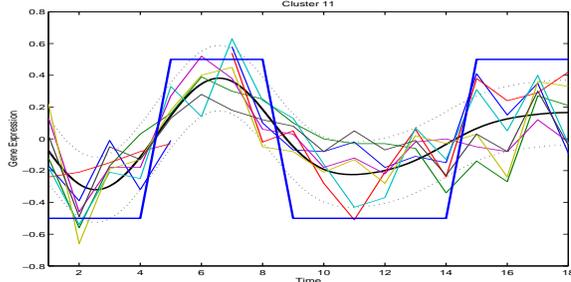


Figure 2. Discretized meta-gene for a gene group from Figure (1).

After discretizing the meta-genes, we infer a Boolean network. The inference algorithm is based on previous work in chemical reaction network generation [7] and contains routines to count, enumerate, and sample Boolean networks that match the clustered and discretized expression profiles. The inference routines run in $O(2^k n^{k+1})$ time, where n is the number of meta-genes available, and k is the maximum connectivity of a given gene.

In order to more easily interpret the results of our Boolean network inference algorithm, we exploit available tools for electronic circuit analysis. In particular, we perform a two-level Boolean minimization on the truth table representation of the inferred gene network using *Espresso*, a well-known logic simplification tool available from www-cad.eecs.Berkeley.edu. *Espresso* produces a minimized truth table for each meta-gene. Since each meta-gene is processed in the same manner, we get a minimized representation of the entire network. This new version of the network simplifies the biological analysis and interpretation.

3. Stability Assessment

Even though the number of possible logic clauses per meta-gene is limited, a large number of possible networks that can be inferred from the same meta-genes. To explore the distribution of possible networks, we expand our logic clause calculation to a set of 1000 randomly sampled networks. We use this calculation to generate statis-

tics which identify the most reliable meta-genes and associated clauses.

We also cluster the sampled networks according to their dynamics. Briefly, we cluster two networks when one network differs from another by a pre-defined hamming distance, as measured using its dynamic expression profile. In other words, two networks having different topologies are clustered if they have similar dynamics. Tests on random networks with different sizes and hamming distance thresholds indicated that for a number of unclustered networks (ranging between 1 and 3^{10} nodes), the number of clusters was no greater than 500.

Finally, we simulate our inferred network using a continuous model called *BioXyce*, which is a parallel electric circuit simulation tool adapted to biological problems [15]. Results are comparable to the original discretized signal. We note that the simulation was not possible using traditional CMOS-based Boolean logic, but we found that a non-CMOS based logic was successful [16].

4. Network Visualization

After the network has been inferred, converted into a minimal set of logical clauses, and been assessed for quality, we present the results in a format amenable to interactive viewing. First, we draw the network using the *dot* graph drawing tool [9], as shown in Figure (3). This tool was programmed to use various colors and shapes to encode information specific to the particular application.

To make the drawing interactive, we displayed it using a web-browser, where each meta-gene is hot-linked and has mouse-over capability. In particular, clicking a meta-gene opens a spreadsheet containing the annotation for the genes in that group, and when a meta-gene is under the mouse, a window pops up to show the original gene expression patterns and corresponding discretization, as shown in Figure (2).

5. Results

We have applied our method to the publicly available yeast time series microarray data in [18]. The steps in the process have been illustrated in Figures (1-3).

In Figure (1), we used the clustering of the time series data previously performed in [20], along with the partitioning by k -means. In this case, we used $k = 100$, and discarded clusters with fewer than 20 genes, leaving 81 meta-genes.

In Figure (2), we used Support Vector Regression with a Gaussian kernel ($\gamma = 2$) and an ϵ -tube width of one and a half times the average standard deviation of the expression values at each time point.

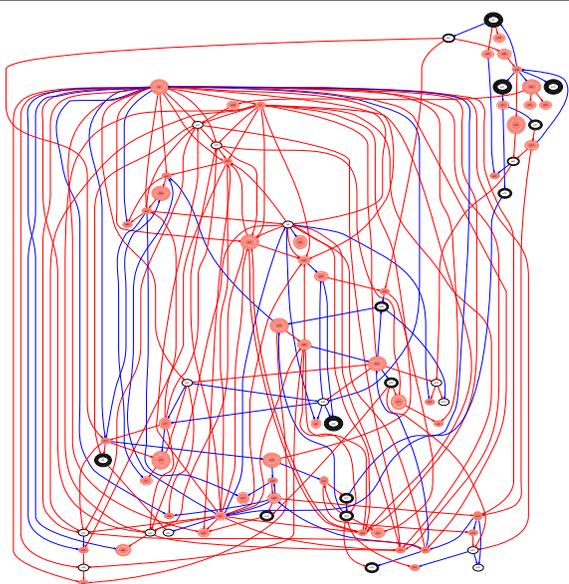


Figure 3. Visualization of final network using the yeast time series data from [18].

In Figure (3), we used different color lines for inhibitory and activation connections, and different color nodes for essential genes. We used circular nodes for genes involved in the cell-cycle, oval nodes for gene not involved in the cell-cycle, and circles around a node to indicate confidence in the relationships for that node. We computed the confidence bands for a given meta-gene in the network using the cumulative distribution of logical clauses from 1000 networks. We found that 14% of the activation/inhibition clauses appeared in all networks, while 45% of the clauses were present in half of the networks. This result indicates that even while a large number of networks can be inferred, there is some consistency across networks.

Finally, the real proof that our method is useful must come from the analysis and interpretation of the final network. Working with our biological collaborators, we have developed two testable hypothesis based on our proposed network. First, we discovered that the meta-gene module in the upper right corner of Figure (3) consists almost entirely of genes involved in exit from alpha-arrest. These cells were exposed to alpha mating factor, which stops the cell-cycle at a checkpoint until it is removed, thereby providing a way to synchronize the cells in the growth medium. The gene groups in the upper right of the drawing seem to be involved in this synchronization process.

Second, we noticed that many of the links in the drawing are inhibitory. This unexpectedly large number of inhibitory controls goes counter to the currently accepted regulatory model and may suggest that genetic networks are more

tightly controlled than has been previously assumed. Further experiments, both laboratory and computational, will be necessary to test these hypotheses.

6. Future Work

We have two primary objectives for the immediate future. First, we have already starting analyzing the stability of our methods in greater detail. In particular, the circles around the nodes in Figure (3) are meant to give an indication of likelihood that a given meta-gene will have the same relationships to other meta-genes in alternate networks generated by the network inference algorithm. We plan to make these computations much more robust by using bootstrapping methods [5] to assess the variance caused by changes in our sampling algorithms. These changes include altering the curve-fitting and discretization parameters as well as considering even more alternate inferences provided by the network inference algorithm.

Second, we intend to perform a full and thorough analysis of time series microarray data that has been collected by a collaborator (A. Martino) in order to infer T-cell regulatory networks. In particular, we will study T-cell regulatory networks triggered through tyrosine kinase receptor activation.

7. Conclusions

The development of this network and visualization environment has required the collaboration of researchers in math (JLF, SM), computer sciences (GD, EM), and yeast genomics (MWW). From the beginning we have focused on *the entire network inference process*. We have developed clustering, discretization, and inference algorithms, and have attempted to validate their output. Finally, we have presented the results using an interactive network browser for accessible biological interpretation. Although we will continue to improve our process, it has already yielded two testable biological hypotheses, one concerning exit from arrested states, and one concerning the level of control present in genetic networks.

8. Acknowledgements

This work was funded by Sandia Laboratory Directed Research and Development project 52533. Some of the related work was funded by the US Department of Energy's Genomics: GTL program (www.doenomestolife.org) under project, "Carbon Sequestration in *Synechococcus Sp.*: From Molecular Machines to Hierarchical Modeling," (www.genomes-to-life.org). Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed

Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

- [1] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Pacific Symposium on Biocomputing*, volume 4, pages 17–28, 1999.
- [2] G. Davidson, B. N. Wylie, and K. W. Boyack. Cluster stability and the use of noise in interpretation of clustering. In *IEEE Symposium on Information Visualization (INFOVIS)*, 2001.
- [3] H. de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.*, 9(1):67–103, 2002.
- [4] P. D'haeseleer, S. Liang, and R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000.
- [5] B. Efron. Bootstrap methods: Another look at the jackknife. *Ann. Statistics*, 7(37):1–26, 1979.
- [6] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression data. *Proc. Nat. Acad. Sci.*, 95(25):14863–14868, 1998.
- [7] J.-L. Faulon and A. G. Sault. Stochastic generator of chemical structure. 3. reaction network generation. *J. Chem. Inf. Comput. Sci.*, 41(4):894–908, 2001.
- [8] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *J. Comput. Biol.*, 7:601–620, 2000.
- [9] E. R. Gansner, E. Koutsofios, S. C. North, and K. P. Vo. A technique for drawing directed graphs. *IEEE Trans. on Soft. Eng.*, 19(3):214–230, 1993.
- [10] J. Goutsias and S. Kim. A nonlinear discrete dynamical model for transcriptional regulation: Construction and properties. *Biophys. J.*, 86:1922–1945, 2004.
- [11] D. Husmeier. Reverse engineering of genetic networks with bayesian networks. *Biochem. Soc. Trans.*, 31:1516–1518, 2003.
- [12] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York, 1993.
- [13] S. Kim et al. A gene expression map for *c. elegans*. *Science*, 293:2087–2093, 2001.
- [14] J. C. Liao et al. Network component analysis: reconstruction of regulatory signals in biological systems. *Proc. Nat. Acad. Sci.*, 100(26):15522–15527, 2003.
- [15] E. E. May and R. Schiek. Simulating regulatory networks using *xyce*. In *Fourth International Conference on Systems Biology*, 2003.
- [16] H. H. McAdams and A. Arkin. Simulation of prokaryotic genetic circuits. *Annual Review of Biophysics and Biomolecular Structure*, 27:199–224, 1998.
- [17] A. J. Smola and B. Scholkopf. A tutorial on support vector regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
- [18] P. T. Spellman et al. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.
- [19] J. Tegner, M. K. Yeung, J. Hasty, and J. Collins. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proc. Nat. Acad. Sci.*, 100(10):5944–5949, 2003.
- [20] M. Werner-Washburne et al. Concurrent analysis of multiple genome-scale datasets. *Genome Research*, 2002.
- [21] M. K. Yeung, J. Tegner, and J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Nat. Acad. Sci.*, 99(9):6163–6168, 2002.