Inferring Genetic Networks from Microarray Data

Shawn Martin[1], George Davidson[1], Eleobeoba May[1], Margaret Werner-Washburne[2], and Jean-Loup Faulon[1]

[1] Sandia National Laboratories, Albuquerque, NM 87185

[2] University of New Mexico, Department of Biology, Albuquerque, NM 87131

Introduction

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.

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 microarray (Davidson *et al*., 2001; Kim *et al*., 2001). 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. 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 this process.

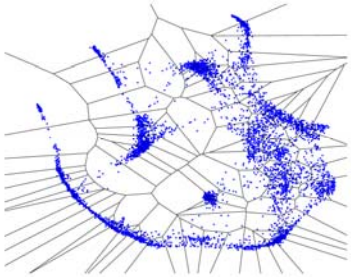

Figure 1. Gene map partitioned with *k*-means.

Since our network inference algorithm is Boolean, we must first discretize the expression levels of our meta-genes. This discretization is accomplished in two steps. First, Support Vector Regression (Smola & Scholkopf, 1998) is used to obtain a single continuous curve representing each meta-gene. Next, an on/off expression profile is obtained by thesholding the resulting continuous curve.

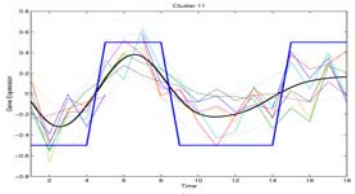

Figure 2. Discretized meta-gene for gene group.

After discretizing the meta-genes, we infer a Boolean network. The inference algorithm is based on previous work in chemical reaction network generation (Faulon & Sault, 2001) and contains routines to count, enumerate, and sample Boolean networks matching 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. This algorithm is shown in Figure (3).

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.

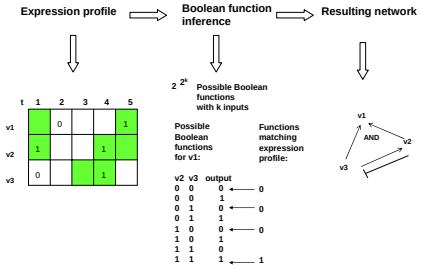

Figure 3. Boolean Inference Algorithm.

Stability Assessment

Even though the number of possible logic clauses per meta-gene is limited, a large number of possible networks 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 statistics which identify the most reliable meta-genes and associated clauses.

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 (part of the GraphViz package available at www.research.att.com/sw/tools/graphviz). This tool was programmed to use various colors and shapes to encode information specific to the particular application. We show a network drawn with *dot* in Figure (4).

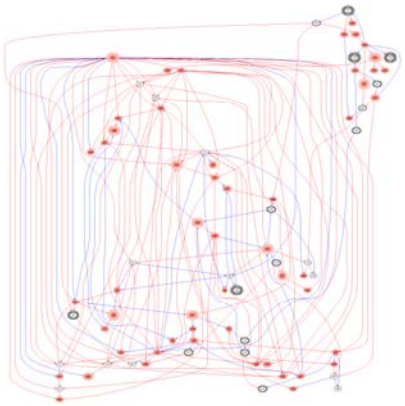

Figure 4. Network visualization for yeast data.

To make the drawing interactive, we display 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 (5).

Results

We have applied our method to the publicly available yeast time series microarray data in (Spellman *et al*., 1998), as shown in the previous figures. For this dataset, we used the clustering of the time series data previously performed in (Werner-Washburne *et al*., 2002) along with partitioning by *k*-means. In this case, we used k = 100, and discarded clusters with fewer than 20 genes, leaving 81 meta-genes.

Next 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.

For the network drawing, we used different color lines for inhibition 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.

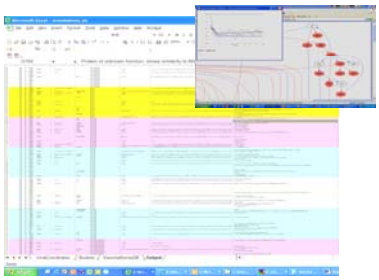

Figure 5. Interactive network visualization environment.

Future Work

A principal objective for future work is the analysis of the stability of our methods in greater detail. In particular, the circles around the nodes in Figure (4) 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 (Efron, 1979) 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. The proposed process is shown in Figure (6).

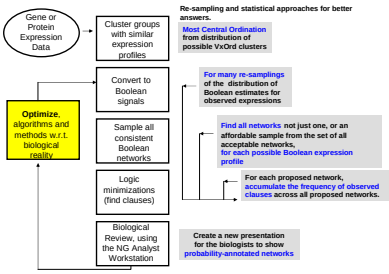

Figure 6. Bootstrapping process for stability assessment.

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.

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.doegenomestolife.org) under project, "Carbon Sequestration in *Synechococcus Sp.*: From Molecular Machines to Hierarchical Modeling," (www.genomes-to-life.org).