Increasing the efficiency of fuzzy logic-based gene expression data analysis

Habtom Ressom, Robert Reynolds, and Rency S. Varghese

Intelligent Systems Laboratory, Department of Electrical and Computer Engineering, University of Maine, Orono, Maine 04469

Submitted 6 August 2002; accepted in final form 14 February 2003

Ressom, Habtom, Robert Reynolds, and Rency S. Varghese. Increasing the efficiency of fuzzy logic-based gene expression data analysis. Physiol Genomics 13: 107–117, 2003. First published February 20, 2003; 10.1152/physiolgenomics.00097.2002. —DNA microarray technology can accommodate a multifaceted analysis of the expression of genes in an organism. The wealth of spatiotemporal data generated by this technology allows researchers to potentially reverse engineer a particular genetic network. “Fuzzy logic” has been proposed as a method to analyze the relationships between genes and help decipher a genetic network. This method can identify interacting genes that fit a known “fuzzy” model of gene interaction by testing all combinations of gene expression profiles. This paper introduces improvements made over previous fuzzy gene regulatory models in terms of computation time and robustness to noise. Improvement in computation time is achieved by using a cluster analysis as a preprocessing method to reduce the total number of gene combinations analyzed. This approach speeds up the algorithm by a factor of 50% with minimal effect on the results. The model's sensitivity to noise is reduced by implementing appropriate methods of “fuzzy rule aggregation” and “conjunction” that produce reliable results in the face of minor changes in model input.

Through the use of DNA microarray technology, time series data for gene expression can easily be prepared on the genome-wide scale, allowing the transcription levels of many genes to be measured simultaneously. With such data, one can attempt to reverse engineer a network of gene interaction. The benefits of characterizing gene interaction are many, for example, the effects of drugs on a regulatory pathway can be characterized; tumor development in cells can be tracked, etc. However, there are many problems that impede understanding gene interaction, such as identifying synchronization, working with low expression levels, the existence of splice variants, etc., of which two are considered here. First, we consider the significant amount of time that is required because of the large volume of data; the complexity of a regulatory network model increases as the number of genes used in the model increases, and as a result this may take a long time to examine. Second, we discuss the fact that any attempt to model or analyze DNA microarray data is likely to be affected by measurement error.

Several methods have been proposed to develop maps of gene interaction, including linear equations (6, 31), differential equations (2), Boolean networks (15, 21), and fuzzy cognitive maps (7). Woolf and Wang (33) introduced an approach based on fuzzy rules of known activator/repressor relationships of gene interaction. Using a normalized subset of Saccharomyces cerevisiae data (3), they applied every possible combination of activators and repressors for each gene. The output from the model was compared with the expression levels of the genes. Gene combinations were ranked based upon the error between the predicted and actual target gene. Those combinations of genes that had a low error and are characterized by the fuzzy rule base are the most likely to exhibit an activator/repressor relationship.

The method of Woolf and Wang (33) attempts to simulate what a human would do in comparing expression levels of genes to find underlying relationships. Different fuzzy models can be developed with the method for different models of interaction, including coactivators and corepressors, as well as the presence of other factors in the cell, such as proteins or other factors necessary for transcription. The method is intuitive, and the results of Woolf and Wang are consistent with the literature on genetic networks of S. cerevisiae. The model itself is an interesting generalization of Boolean networks, where genes are not either “on” or “off” but are often both “on” and “off” at the same time.

Although this method appears to be effective, a few drawbacks exist. First, a significant amount of time is required to run the algorithm. Since every triplet of genes (one as the activator, one as the repressor, and one as the target gene) is checked, the algorithmic complexity is $O(N^3)$, where $N$ is the number of genes. As a result, examining a large number of genes takes a long time. Second, the algorithm will not scale well to more complex models. A model that includes coactivators and corepressors (4 inputs) would have an $O(N^5)$ complexity; the time required to run such an algorithm would be on the scale of years instead of hours for large $N$. Woolf and Wang (33) note that their algorithm, which was written in C programming language and...
run on an 8-processor SGI Origin 2000 system, took 200 hours to analyze the relationships between every triplet in 1,898 genes. By making minor changes to their source code and running it on a Pentium II system, we were able to reduce the amount of time required to ~25 hours.

Computational time is a significant obstacle in developing more complex fuzzy models. A solution to the problem may lie in preprocessing the data. If it is possible to find triplets of genes that are unlikely to fit the model before running the algorithm, then the unlikely triplets can be ignored when the algorithm is run, thus saving an amount of time approximately proportional to the number of gene triplets not considered at later stages. To address the issue of robustness, two approaches can be considered. The first approach is to improve the microarray technology to increase signal-to-noise ratio. Analysis of variance approaches (12) may also provide more accurate readings and knowledge of signal-to-noise ratios. However, issues such as the stochastic nature of gene expression may not be eliminated, and some degree of error will have to be dealt with. The second approach is to use models that are more resilient to minor variations in the expression data.

In this paper we propose a preprocessing step to reduce computation time and a new fuzzy methodology to improve the robustness of gene regulatory network models. The improved algorithm achieves a speed up of 50% and significant resistance to noise in expression profiles. Since the fuzzy logic-based model deals with qualitative descriptors (such as “high” or “low”) rather than actual expression levels, it is able to deal with imprecision or low levels of noise with minimal changes to the resulting models.

One method of preprocessing is data clustering. To analyze gene expression data, a wide range of clustering algorithms have been proposed in the literature. Examples include hierarchical clustering (8), adaptive resonance theory (ART) (30), self-organizing maps (SOM) (23), k-means (24), graph-theoretic approaches (1, 11), the gap statistic (25), fuzzy ART (26), fuzzy c-means (10), fuzzy Kohonen methodology (9), and growing cell structures network (9). Mangiameli et al. (17) evaluate the performance of several different clustering algorithms like SOM and hierarchical clustering using artificial data. They conclude that SOM is superior in both accuracy and robustness.

SOM is a neural network that provides a mapping from a multidimensional data to a discrete one- or two-dimensional space. The method is robust, scalable, flexible, and reasonably fast. In SOM, nodes can be connected in either a one- or two-dimensional network. Each node is associated with a multidimensional weight vector, which corresponds to the center of the cluster that the node represents. The weight vectors are randomly initialized and are updated during a learning process based on the features from the input space. An input vector is selected from the input space, and the node whose weight vector has the minimum Euclidean distance to this input vector is identified. This node is referred to as a “winning node.” The weight vector of the winning node is adjusted so that it is more similar to the input vector. Also, neighboring nodes are updated to a degree proportional to their distance from the winning node in the network topology. This weight-updating process is repeated until no noticeable change in the weight vectors is observed. When the learning process is complete, the final weight vector of each node represents a cluster center. Moreover, neighboring clusters will have similar cluster profiles, while more distant clusters become increasingly diverse (13).

In our experiments, we clustered datasets gathered from yeast data (3), rat CNS data (32), and yeast elutriation and cdc15 data (22) using SOM with varying numbers of nodes. We ran each combination of three cluster centers representing activator, repressor, and target through Woolf and Wang’s algorithm and ranked them according to how well each fit the fuzzy model. Only gene triplets whose corresponding cluster center test statistics surpassed a specified cutoff were examined; i.e., if a gene triplet’s corresponding cluster centers do not fit the model well, it is unlikely that the triplet will fit model of activator/repressor. The experiment was performed on the four different public databases with varying numbers of clusters in the SOM as well as different cutoff limits. In this paper we show that with a sufficient number of clusters and a well-chosen cutoff, the time required by the algorithm can easily be reduced up to 50% or more with minimal effect on the algorithm’s reliability.

There are many potential causes for noise in gene expression data, most originating from the stochastic nature of gene interactions and microarray technology (27). Attempting to create a model from data that is corrupted by such a high noise is extremely difficult. To make a fuzzy model less susceptible to such noise, we explored methods of conjunction and rule aggregation that appear to produce reliable results in the face of minor changes in model input (20). The hybrid model combines attributes of the model of Mamdani and Assilian (16) and standard additive model (SAM) of Kosko (14). We show that the use of the standard Mamdani model can improve the performance of Woolf and Wang’s fuzzy algorithm by being more robust.

SYSTEMS AND METHODS

Fuzzy logic is a generalization of Boolean logic in which the truth value of a statement can be in the continuous interval between 1 (completely true) and 0 (completely false). Fuzzy logic can deal with the subjectivity of qualitative measurements. Using membership functions, one can establish a qualitative set of rules that can model a control system or a physical process. The rules are expressed in qualitative values that a human can easily understand. For example, one might say, “for a gene that is regulated by an activator and repressor, if the activator’s expression level is high and the repressor’s expression level is low, then the corresponding gene’s expression will be high.” The
membership values for all inputs can be applied to all the rules to obtain a fuzzy output that can be converted back (defuzzification) to a quantitative value (in this case, the gene's expression level) through the membership functions. Fuzzy logic can thus provide a quickly implemented and easy to understand nonlinear approximation of a process.

**Woolf and Wang's Algorithm**

Woolf and Wang (33) used the yeast data from the *S. cerevisiae* cell cycle expression database (3), which has 6,321 genes at seventeen 10-min samples. From this data, they selected 1,898 genes that satisfied two conditions (33). 1) The noise level had to be 30 in fluorescence intensity; thus the minimum expression level was set to be greater than 30 to filter out the nondetectable genes in the samples. And 2) the ratio between the maximum and minimum expression had to be greater than 3 to ensure that the observed signal change was significant. Woolf and Wang's algorithm applied all combinations of the 1,898 genes to a fuzzy model with membership functions and the rule base decision matrix as shown in Fig. 1.

The “goodness of fit” of the model for a gene triplet was evaluated using two scores. The first score was the square of the error between the model’s output and the target gene’s actual expression level. The second score was the statistical variance between the total firing of the fuzzy rules in each box on the decision matrix. These scores were calculated and stored for each gene triplet. In order for a gene triplet to be counted as a significant result, an error cutoff of less than 3% and variance of the firing of fuzzy rules of less than 1.5 had to be met. The reason for the error cutoff is obvious: model outputs with a high error indicate that the gene triplet probably does not exhibit the activator-repressor-target relationship. The reason for the variance criterion may not be as obvious. A low variance implies that all of the rules were used about equally in generating the model output, meaning that the model’s entire range had been tested. If the output of the model had a low error and had been tested for the model’s entire range, then the results are more likely to be credible. High variance implies that not all of the rules were used equally and parts of the model’s range have not been tested. Thus high variance results are unlikely to be credible.

Woolf and Wang also proposed that the fuzzy model could be expanded to more complex models than the activator-repressor-target gene model. Models that incorporate coactivators and corepressors can be used, as well as models that incorporate the concentration of proteins or other compounds such as Ca$^{2+}$ that are important for gene expression and transcription. One only needs to design a proper rule base to model any form of gene interaction. Each additional gene input to the model increases the algorithmic complexity, thus dramatically increasing the time required to run the algorithm.

Woolf and Wang’s fuzzy logic analyses were written in C programming language and run on an 8-processor SGI Origin 2000 system. It required 200 hours to analyze the relationships between the 1,898 genes.

**Clustering to Improve Run Time**

A first step toward the rapid and comprehensive interpretation of the data is the clustering of the genes with respect to the expression. Clustering is routinely used to find genes with similar expression profiles. We can attempt to use gene clusters as metadata for gene expression datasets. If a particular combination of clusters does not fit the model well, then it is unlikely that any genes with similar expression profiles will fit the model well. This can be shown through an analysis of how the data is processed.

In the mathematical formalization of our approach (18), each gene is represented as a vector of time series data, which is normalized between 0 and 1. Let $X$ be a gene expression matrix containing gene expression profiles (in our experiments, $X$ consists of two vectors representing two expression profiles, i.e., activator and repressor), and let $y$ be the output of the model, we define $y = f(X)$, where $f$ is a vector-valued function representing the model’s transfer function. The output of the model, $y$, represents the ideal expression profile of the target gene. Let $z$ be a vector representing the actual expression profile of the target gene, then the
mean squared error (MSE) between the model output and the expression profile of the target gene is given by

\[
\text{MSE}(X, z) = \frac{1}{N} \langle y - z \rangle^T (y - z) 
\]

\[
= \frac{1}{N} \langle f(X) - z \rangle^T (f(X) - z) 
\]

where \( N \) is the vector length (i.e., number of points in the time series); \((y - z)^T\) denotes the transpose of the vector \((y - z)\). Now, assume that \( X, z \) are metadata for \( X \) and \( z \), respectively. That is, \( X \) and \( z \) contain general information about the expression level that could be provided by using cluster centers of the clusters closest to \( X \) and \( z \). Defining \( \delta X, \delta y, \) and \( \delta z \) as

\[
\delta X = X - \bar{X} \quad \text{(2)}
\]

\[
f(X) = f(X) - \delta y \quad \text{(3)}
\]

\[
\delta z = z - \bar{z} \quad \text{(4)}
\]

we denote the difference between the MSE of the model for a given gene expression matrix \( X \) and the model's MSE for the corresponding metadata \( X \) by \( \delta \text{MSE} \), which is given by

\[
\delta \text{MSE} = \text{MSE}(X, z) - \text{MSE}(\bar{X}, \bar{z}) \quad \text{(5)}
\]

\[
\delta \text{MSE} = \frac{1}{N} \langle (f(X) - z) \rangle^T (f(X) - z)
\]

\[
- (f(X) - \bar{z})^T (f(X) - \bar{z}) \quad \text{(6)}
\]

Substituting Eqs. 3 and 4 in 6, we get

\[
\delta \text{MSE} = \frac{1}{N} \langle (f(X) - z) \rangle^T (f(X) - z)
\]

\[
- (f(X) - \delta y - z + \delta z)^T
\]

\[
\times (f(X) - \delta y - z + \delta z) \quad \text{(7)}
\]

If the dataset is amenable to clustering (i.e., the majority of gene expression profiles would be close to a cluster center), then we can assume that the difference between \( X \) and \( \bar{X} \), as well as between \( z \) and \( \bar{z} \) is close to 0, i.e.

\[
\delta X = X - \bar{X} \to 0 \quad \text{(8)}
\]

\[
\delta z = z - \bar{z} \to 0 \quad \text{(9)}
\]

If we assume that \( \delta X \) is small around most input values of \( X \) and that the gradient of the output space \( y \) is relatively small (which is the case for most \( X \))

\[
\delta y = f(X) - f(X) \to 0 \quad \text{(10)}
\]

Hence, as \( \delta z \) and \( \delta y \) go to 0

\[
\delta \text{MSE} = \frac{1}{N} \langle (f(X) - z) \rangle^T (f(X) - z)
\]

\[
- (f(X) - \delta y - z + \delta z)^T
\]

\[
\times (f(X) - \delta y - z + \delta z) \to 0 \quad \text{(11)}
\]

Therefore, considering that we can cluster the data so that most of the expression profiles are relatively close to their cluster centers, cluster centers and their corresponding gene profiles will be similar and the difference in the MSE will be minimal. Thus, if a combination of cluster centers does not fit the model well, then genes close to those cluster centers will not fit the model well. With prior knowledge of how cluster centers fit the model, we can eliminate combinations of genes whose nearest cluster centers do not fit the model well. Thus, by not analyzing those combinations in the fuzzy model, we can save tremendous computation time.

Hence, clustering allows us to effectively reduce the dataset to a handful of time series that approximately represent the data as a whole. Provided that the standard deviation between genes and their corresponding cluster center is low, we can assert that the cluster center’s expression profile is approximately equal to the genes in the cluster.

To illustrate the use of clustering in modeling a gene regulatory network, gene expression profiles representing cluster centers grouped into two sets of triplets are shown in Fig. 2. One can easily determine whether large groups of genes, represented by these cluster centers, are likely to fit the model. For example, from Fig. 2 (top), we can see that an increasing activator and decreasing repressor would cause the target gene to increase quickly. These cluster triplets make sense intuitively and should be included in analysis. Figure 2 (bottom) shows counterexamples to the combinations shown on the top. These cluster triplets do not make sense intuitively and should not be included in analysis.

To find which cluster combination of three centers make sense intuitively, we can use the fuzzy algorithm. If a combination of three cluster centers does not fit the fuzzy model well, then the corresponding gene triplets will be ignored from being tested for model fitness. The ignored gene triplets are unlikely to fit the model well and would probably not meet the error cutoff used to determine whether or not a gene triplet is significant.

In our experiments, a digital low-pass filter was used to eliminate noise and extract general expression trends for filtering. The filtered data was clustered with SOM using GeneCluster (23). Several SOM were trained for each dataset with different numbers of clusters.

---

**Fig. 2.** Cluster triplets that would fit the model well (top), and cluster triplets that would not fit the model well (bottom).
The Improved Algorithm

In this section, we describe the steps involved in the improved version of Woolf and Wang’s algorithm to analyze gene expression data. First, the algorithm is run with no error or variance cutoffs using the cluster centers for triplets. Each triplet of cluster centers is ranked according to its error in relation to the model. The algorithm is then run on the genes themselves. Before gene analysis, a file containing all cluster triplets and their error in relation to the model is read. The cluster triplets are separated based upon the target cluster. The centers of the clusters they belong to are evaluated in one of the following two ways: the percentile ranking method and the error threshold method.

1. Percentile ranking method. The corresponding cluster triplet must be above a certain ranking percentile for the cluster of the target gene. Ranking is determined by the model error for cluster triplets; lower error implies a higher rank.

2. Error threshold method. The corresponding cluster combination must have an error score below a previously specified threshold.

If the gene triplet’s corresponding cluster triplet does not meet the specified threshold, then the gene triplet is not analyzed, and the algorithm proceeds to the next triplet.

A percentile specified at runtime is used to separate the triplets for each target cluster into triplets to check (those that make the percentile cutoff in terms of low error) and triplets to ignore (those that do not make the percentile cutoff in terms of low error). Hence, for a gene triplet to be considered as a potential combination, the corresponding cluster triplet must be above a certain ranking percentile for the cluster of the target gene. Ranking is determined by the model error for cluster triplets; lower error implies a higher rank. The reason that the percentile cutoff is applied to cluster triplets separated by cluster instead of the global set of triplets is to allow target genes from all clusters to be considered. For example, a particular target cluster may have corresponding activator and repressor targets with low error, while another target cluster may have corresponding activator and repressor clusters with only slightly higher error in relation to the first set; the highest ranking triplet in the second group has an error slightly higher than the highest ranking triplet in the first, etc. If a global percentile cutoff was used, then the combinations from the second cluster may all be ignored, even though it is likely to have many valid gene triplets. Separating the cluster triplets by target cluster allows more genes to be considered as targets while still eliminating unlikely gene triplets from analysis.

For the error threshold method, the idea is to use a cluster score threshold to select the cluster nodes whose corresponding genes would be analyzed. It is assumed that there exists some function \( g(h) \) where \( h \) is a maximum desired error score for gene combinations and \( g(h) \) is the corresponding minimum error threshold for the corresponding cluster combinations.

The choice of optimal score threshold should be dataset independent; it should only be a function of the model itself and the error cutoff set for gene combinations.

Before applying and testing the fuzzy model, the algorithm checks the clusters that the activator, repressor, and target represent. If the cluster combination is not marked, then the gene combination is not likely to fit the model, and the algorithm skips over it. Otherwise, operation continues as in the original algorithm; the model is applied, and the error and variance of the model output are checked. The number of gene triplets that pass the error and variance cutoffs as well as the number of gene triplets analyzed (and ignored) and the time required running the algorithms are stored. The algorithm was written in ANSI C and compiled and run on clusters of Pentium II PCs running Red Hat Linux 7.0. The source code is freely available at http://www.intsys.maine.edu/fuzzygene.htm.

Improving Resilience of the Fuzzy Model

For a given set of membership functions and fuzzy rules, fuzzy models can differ in many ways, including fuzzy conjunction (AND) operations, rule aggregation, and defuzzification. Woolf’s algorithm uses addition for fuzzy “AND” operations, averaging for rule aggregation, and a modified centroid method for defuzzification. As will be shown, this method produces an unusual output space with sharp gradients in several regions. We decided to use a few different methods and analyze their output and reliability. We used Mamdani’s model (16), Kosko’s SAM (14), and a hybrid model that attempts to take the best attributes of the Mamdani and SAM models.

Mamdani’s model is a classic model that uses the drastic product (i.e., minimum) operation for conjunction and a drastic sum (i.e., maximum) operator for rule aggregation. The model does not provide a specific method for defuzzification; it is up to the model designer to decide the method, which can include mean of median (MOM), center of area (COA or centroid), or any other method. A minimum operator on fuzzy inputs makes intuitive sense for gene interaction; the truth value of a particular rule is going to be bound by the minimally expressed gene. For example, if an activator’s expression level is mostly MED and a little HIGH, while a repressor’s expression level is mostly LOW and a little MED, the rule “if activator is HIGH and repressor is LOW, then target is HIGH” should be limited completely by the fact that the activator is not particularly HIGH.

Kosko’s SAM uses a product operation for conjunction, a sum operation for aggregation, and the centroid method for defuzzification. Centroid defuzzification is performed by scaling membership functions instead of clipping them at the level of rule application. The hybrid model combines attributes of the Mamdani model and SAM. It uses a product operation for conjunction, a maximum operator for aggregation and a centroid defuzzification that involves scaling as in SAM.
To analyze the output space of each method, we calculated the output of each fuzzy system when presented with all combinations of activator and repressor levels in increments of 0.01. The gradient was calculated using the output surface. The mean and standard deviation of the gradient matrices for each method was calculated.

To analyze the effect of noise on the fuzzy models, the output of the algorithm in Ref. 33 with the data from Ref. 3 was obtained for each method. The output includes all gene triplets that fit the model well. A Monte Carlo simulation was run on the gene triplets that fit each model well by distorting each time point by a random noise percentage and analyzing the result on the model output. Each triplet was rerun 20,000 times with maximum distortion set at different levels from 5–30%. The mean and standard deviation of the model MSE for the 20,000 experiments were stored. Each triplet's average MSE was plotted vs. its original MSE. Similar plots were made for the standard deviation of MSE for each time point. Linear regressions were found for the average error graphs. If two model types produce similar MSE regression lines with slope near 1, then the one with the smaller standard deviation was considered to be less sensitive to noise; since two regression lines with the same equation can have different standard deviation graphs.

Sensitivity to noise can be related to the regression line: a model is generally less sensitive to noise if the slope of the regression line is close to 1 and the y-intercept is close to 0 (i.e., no average change in MSE due to noise). A slope of 1 would imply that on average, all model outputs would be distorted by the same factor, regardless of the original MSE. Minimizing the y-intercept value is also of interest, but is not necessary for proper operation: if we know that all error scores are offset by the same value due to error, we can simply raise our error cutoffs to get the same results.

Model validation was performed in a manner similar to Woolf and Wang's method. The best scoring triplets were checked if they exist in a list of known activator/repressor complexes, which we obtained from Refs. 4
and 5. We searched for 45 known transcription factors in the yeast genome and compared the number of times they appeared in the results to how many of them are present in the dataset; transcription factors directly affect gene expression, so they should be present in a high percentage of the results. We searched for the most frequently appearing gene pairs in the triplets to see whether they exhibit known biological relationships.

EXPERIMENTAL TESTS AND RESULTS

Improved Run Time

Gene expression data was gathered from yeast data (3), rat CNS data (32), and yeast elutriation and cdc15 data (22). The data were run through a filter that eliminated genes whose maximum expression was below a noise threshold (set at 30 for yeast data and its equivalent for rat CNS data) or whose difference between the maximum and minimum expression level was less than a factor of 3.

Each of the datasets was analyzed assuming differing numbers of clusters and with 50%, 67%, 75%, and 100% of the cluster combinations kept. The time required and triplets found for each test case were compared against the Woolf and Wang algorithm (i.e., 100% cluster combinations saved). Our results for the yeast cdc15 dataset are shown in Fig. 3. Similar results are obtained for other datasets including yeast data, rat CNS data, and yeast elutriation data (18). Since the improved algorithm we propose in this paper does not interfere with the fuzzy models themselves, the results from each test run is a subset of the results of the Woolf and Wang algorithm. Analysis of the results proved this claim. Figure 3 shows the percentage of original result expressed for a given number of clusters and percentage of cluster triplets kept.

From the results, we can see several definite trends. Increasing the number of clusters in the SOM increases the percentage of results obtained. However, there is a point where the change in percentage of results obtained by increasing the number of clusters (marginal return) approaches 0. From the empirical results of the four datasets, it appears that the point where marginal returns diminish is approximately

Fig. 4. Output surfaces of four fuzzy models (clockwise, from top left): Woolf and Wang, Mamdani, hybrid, and SAM.
equal to \( \frac{1}{2} \) the number of time points in the dataset. Increasing the number of clusters does little, if anything, to affect the time required by the algorithm. The time required to run the algorithm is instead proportional to the percentage of cluster triplets kept. Increasing the percentage of cluster triplets kept will increase the percentage of original results found, but there is also a point where the marginal returns diminish to the point where keeping a larger percentage of the cluster combinations does little to improve the percentage of results obtained. Although the point of low returns varies from dataset to dataset, it appears that 67\% is sufficient in most cases.

A disadvantage of selecting percentages of cluster combinations as in percentile cutoff method is that selecting the percentage for optimal time saving and results is completely subjective to the dataset. It was observed that, for a particular percentage of the original (33) results obtained, the error score of the worst combination of clusters that was checked was relatively constant. This fact led to the idea of using cluster error score threshold to select the cluster nodes whose corresponding genes would be analyzed.

To analyze the accuracy of the algorithm in identifying valid gene combinations, we define a “99.9\%” point, which is the lowest cluster error threshold that returns 99.9\% of the results of Woolf and Wang’s algorithm. It appears that, for a given desired error cutoff the 99.9\% point is independent of the dataset or the number of clusters used (provided we examine a near-optimal number of clusters). In all three datasets, the point is approximately at an MSE threshold of 7.5\% for a desired cutoff of 1.5\%, 8\% for a desired cutoff of 2\%, and 8.5\% for a desired cutoff of 2.5\%. For example, our experiment with cdc28 dataset for a cutoff of 2\%, 99.9\% of the triplets obtained by Wolf and Wang were found at a cluster error threshold of 7.5\% for different numbers of clusters ranging from 12 and 15. The amount of time required to reach the 99.9\% point was on the average 67\% of the original time required, i.e., the time required to test all possible triplets. For the range of gene combinations (i.e., those with an MSE of 2.5\% or less), we found that the error threshold is linear and independent of factors other than the desired cutoff. It is also evident that the time required to run the algorithm in this manner is independent of the desired error cutoff and is only dependent upon the cluster error threshold and the dataset used.

The cluster error threshold method allows us to have a prior knowledge of optimal conditions and thus be able to realize the benefits of clustering as a preprocessing method. The amount of time saved varies since it becomes dependent on factors such as complexity of the dataset.

### Improved Resilience to Noise

The output spaces for each of the Woolf, Mamdani, SAM, and hybrid models are depicted in Fig. 4. The analysis of the gradient of the output space of each model is presented in Table 1.

The highly irregular response of Woolf’s model is reflected in a high average gradient as well as the high standard deviation: most of the change in output is localized in small areas of the input space. The Mamdani model has a much lower average gradient and standard deviation. The SAM has a higher average gradient, but an extremely low change in standard deviation shows that the model has a more consistent gradient. The hybrid model appears to be a compromise between the Mamdani model and the SAM.

#### Table 1. Gradient analysis of the fuzzy models

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Woolf</th>
<th>Mamdani</th>
<th>SAM</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean gradient</td>
<td>10.31</td>
<td>5.48</td>
<td>6.57</td>
<td>6.02</td>
</tr>
<tr>
<td>SD of gradient</td>
<td>13.68</td>
<td>4.64</td>
<td>0.68</td>
<td>2.88</td>
</tr>
</tbody>
</table>

SD, standard deviation; SAM, Kosko’s standard additive model.

**Fig. 5.** Monte Carlo error simulations for the model of Woolf and Wang: scatter plots of old score vs. average new score (A) and old score vs. standard deviation (B).
Figures 5 and 6 depict results of Monte Carlo error simulations run for 30% noise for Woolf and Mamdani models, respectively. Figures 5A and 6A are scatter plots of old MSE vs. average of new MSE, whereas Figures 5B and 6B depict scatter plots of old MSE vs. standard deviation. Note that MSE and standard deviation are multiplied by a factor 100,000 to obtain a score that is easier to read. The black lines in Figs. 5 and 6 indicate the actual regression line, while the gray lines (A) correspond to the ideal lines. The results for the hybrid and SAM models are not included, as their noise responses perform better than Woolf response but are not as robust as the Mamdani model. A more complete set of error simulation graphs can be found in Ref. 18.

From the graphs, it appears that the Mamdani model produces regression lines with a slope closest to 1 for all potential noise distortions. This implies that, on average, the primary effect of noise on the model is only to add a constant error offset to the noise-free error score. The original error score for the inputs (i.e., the noise-free error score) has little or no effect upon the noise-distorted data's error score. If the standard deviation of noise-distorted error score is also low, as is the case with the Mamdani model, then we can say that the majority of gene input combinations are distorted by approximately the constant error offset. If the dataset's noise interval can be estimated (12), then one could raise the desired error cutoff by the value of the constant error offset to obtain the majority of the genes that are likely to fit the model under noise-free conditions. The other three models (Woolf, SAM, and hybrid) have regression line slopes significantly greater than 1, implying that high-error gene combinations will be distorted by a higher amount in the presence of noise. The standard deviation of scores around the results from Woolf's algorithm is much higher than the other three models.

Transcription factor enrichment results are indicated in Table 2. The percentages are derived from the results of each model with an error score cutoff of MSE of 2% and a variance cutoff of 0.2. The “Ratio of enrichment” is the ratio of percentages of results with transcription factors in them to the percentage of transcription factors in the input set (3.97% of the input genes). All of the models appear to report a disproportionate amount of low-error results containing transcription factors. However, the Mamdani and hybrid models appear to yield a higher percentage of results with transcription factors than Woolf's model or SAM. This may imply that these models are better at extracting gene relationships.

The algorithm's output using the Mamdani model was analyzed and compared with the outputs of Woolf's model. The gene relationships of the HAP1 regulatory network were found to have similar error and variance scores as indicated in Ref. 33. Most of the variance scores were higher, but the variance calculations appear to produce higher variance scores with the Mamdani model in general, so an increased variance score cutoff would eliminate the problem. This shows that the Mamdani model has the ability to find some known relationships. The most common pairs of genes were found and are summarized in Table 3. Most of the gene relationships were obtained from the Proteome YPD database (4).

The common pairs show us that the Mamdani model extracts many correlated pairs of genes. There is no known causal relation between the two, but they ap-

Table 2. Transcription factor enrichment of the fuzzy models

<table>
<thead>
<tr>
<th>Model Type</th>
<th>TF % in results</th>
<th>Ratio of enrichment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woolf</td>
<td>8.96%</td>
<td>2.26</td>
</tr>
<tr>
<td>Mamdani</td>
<td>10.53%</td>
<td>2.66</td>
</tr>
<tr>
<td>SAM</td>
<td>9.43%</td>
<td>2.38</td>
</tr>
<tr>
<td>Hybrid</td>
<td>10.51%</td>
<td>2.65</td>
</tr>
</tbody>
</table>

TF, transcription factor.
To accomplish this, three approaches should be investigated. First, research should be directed to the use of time delayed and instantaneous expression values of activators and repressors as input to the fuzzy model. We believe that this approach can improve the prediction of the expression level of the target gene. Second, coactivators and corepressors should be included in the fuzzy model. This can pave the way toward the creation of a generalized gene regulatory model that can accommodate any number of genes. Finally, more fuzzy sets such as very low, low, medium, medium high, and very high will be added to the fuzzy model to make the model response more continuous. This will, however, require modifying the decision matrix and studying the appropriate fuzzy rules, which will increase the model complexity. Other approaches such as artificial neural networks or adaptive regression schemes can be used to capture the relationships between activators and repressors from the data itself. The performance of these approaches highly depends on the quality of data available. We believe combining these approaches with the fuzzy logic-based analysis would be an interesting future work.

We are indebted to Peter Woolf of the University of Michigan for his help with the original fuzzy algorithm. We thank our colleague Padma Natarajan for insightful comments about this manuscript.
We also thank our colleagues Mohamad Musavi and Cristian Domnisoru for many helpful discussions.

The work was supported in part by the Department of Energy EPSCOR program and by Maine Science and Technology Foundation Award 99-04.

REFERENCES