2001

Gene Expression Data Analysis Using Fuzzy Logic

Robert Reynolds

Gene Expression Data Analysis Using Fuzzy Logic

By

Robert Reynolds

B.S. University of Maine, 2000

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

(in Computer Engineering)

The Graduate School

The University of Maine

December, 2001

Advisory Committee:

Habtom Ressom, Assistant Professor of Electrical and Computer Engineering, Advisor

Mohamad T. Musavi, Professor of Electrical and Computer Engineering

Bruce Segee, Associate Professor of Electrical and Computer Engineering

Keith Hutchison, Professor of Biochemistry, Microbiology, and Molecular Biology

Gene Expression Data Analysis Using Fuzzy Logic

By Robert Reynolds

Thesis Advisor: Dr. Habtom Ressom

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Science
(in Computer Engineering)
December, 2001

DNA microarray technology allows for the parallel analysis of the expression of genes in an organism. The wealth of spatio-temporal data provided by the technology allows us to attempt to reverse engineer the genetic network. Fuzzy logic has been proposed as a method of analyzing the relationships between genes as well as their corresponding proteins. Combinations of genes are entered into a fuzzy model of gene interaction and evaluated on the basis of how well the combination fits the model. Those combinations of genes that fit the model are likely to be related. However, current analysis algorithms are slow and computationally complex, sensitive to noise in gene expression data, and only tested and validated on simple models of gene interaction. This thesis proposes improvements to the fuzzy gene modeling method by reducing the computation time, altering the model to make it more robust with respect to noise, and generalizing the model to accommodate any combination of genes and model of gene interaction. The improved algorithm achieves a speed-up of 15-50%, significant resistance to noise, and a degree of generality that enables the analysis of large gene complexes.

ACKNOWLEDGMENTS

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

I would like to thank all the members of my committee: Dr. Habtom Ressom, my advisor, for all the help and support for the past year of work; Dr. Mohamad Musavi for his aid and advice for publishing papers; Dr. Bruce Segee, who first introduced me to fuzzy logic and taught me a great deal during my undergraduate years; and Dr. Keith Hutchinson for his help with the biological background I needed to complete the research.

I would also like to thank my parents for their years of support and for instilling a strong desire to learn in me, Robert Gaboury for helping me realize my potential and getting me interested in engineering, Jesse Cousins for keeping me sane through the past five years and his help in learning LaTeX, and the members of the University of Maine Marching and Pep Bands (as well as the brothers of Kappa Kappa Psi, National Honorary Band Fraternity) for leaving me with so many great memories of my years here and making sure that I did more than just work.

I would also like to thank Peter Woolf of the University of Michigan for his help in understanding the original fuzzy algorithm and providing help concerning the background of his work, and Steve Cousins for his help in obtaining computational resources for my experiments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER 1

Introduction

1.1 Background

DNA Microarray technology [1] allows us to analyze the relative expression levels of a group of genes of an organism simultaneously. Instead of being forced into only examining a few genes at once, we now have the "whole picture" of the expression of genes. Microarray technology also is relatively fast, allowing the quick creation of spatial data (i.e., expression levels of different cells in an organism at a particular time) as well as temporal data (i.e., expression levels of the same cell population in a time series).

With the new wealth of spatio-temporal data obtained from microarrays, many different methods have been proposed to make sense of the data. Clustering algorithms have been used to group genes by their expression profiles [2, 3] to find related genes. Others have attempted to reverse engineer the network of genetic interactions through methods such as linear matrices [4, 5], series of differential equations [6], and Boolean Networks [7]

Another method that has been attempted is fuzzy logic. Woolf and Wang [8] have developed a fuzzy model of known gene interaction (an activator/repressor relationship in this case) Using a normalized subset of *saccharomyces cervisae* data from [9], they apply every possible combination of activators and repressors for each gene. The output of the model, which is the ideal expression of a gene that is regulated by that activator and repressor, is compared to the expression level of a third gene known as the target gene. Gene combinations are ranked based upon the mean squared error between the model and the target gene and variance between the application of the fuzzy rules over the time period. Those combinations of

genes that have a low error are the most likely to exhibit an activator/repressor relationship.

The method attempts to simulate what a human would do in comparing expression levels of genes to find the underlying relationships. Different fuzzy models can be developed for different models of interaction, including co-activators and co-repressors as well as the presence of other factors in the cell, such as proteins or assorted compounds necessary for transcription. The method is intuitively pleasing and the results are consistent with literature of genetic networks of *saccharomyces cervisae.* 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.

While the method appears to be effective, a few drawbacks exist:

1. The algorithm is of $O(N^3)$ complexity; every triplet of genes (one as the activator, one as the repressor, and one as the target gene) is checked. As a result, large numbers of genes take a long time to examine; the 1891 gene subset used by Woolf and Wang required more than 200 hours on an 8-processor SGI Origin 2000 system. With simple optimizations, the time can be dramatically reduced, but the algorithmic complexity remains. Each input added to the model increases the complexity by an order of magnitude; the time required to analyze a model with two activators and two repressors would be on the scale of years using similar computational resources.

2. Microarray data is inherently noisy; most experiments cite that detectable changes in gene expression are limited to detections of doubled expression or greater; [10] cites a minimum detectable change of 1.8, implying a signal error of 29%. There are many potential causes for noise, including improper binding and the stochastic nature of microarray technology. Attempting to create a model from data that is corrupted by such a high noise margin is

extremely difficult; it is likely that the model developed will not accurately predict proper connections between genes. Improving microarray technology or experimental methods [11] to lower the noise ratio will help reduce model error. However, some issues, such as the process's stochastic nature, may not be eliminated by new technology, and some degree of error will have to be dealt with. Woolf and Wang's original model, as will be shown in Chapter 5, is highly vulnerable to slight changes in the model inputs. Any noise in the data will dramatically affect the results.

3. Woolf and Wang have only analyzed a simple activator-repressor model. More complex models that introduce multiple activators or repressors have not been tested.

1.2 Purpose of the Research

This thesis attempts to improve and expand upon the proof-of-concept algorithm proposed by Woolf and Wang. We propose the following solutions to the problems mentioned in the previous section:

1. We propose the use of clustering gene expression data as a preprocessing method to eliminate combinations of genes that are not likely to fit the model.

2. We propose altering the methods used by Woolf and Wang to conjoin and aggregate fuzzy data to reduce the sensitivity of the model to small variations in the inputs while still producing valid results.

3. We propose a generalized version of Woolf and Wang's fuzzy model to accommodate any number of activators and repressors in the model.

The above improvements will improve the performance, robustness, and generality of the model to make it more viable as a method for the analysis of gene expression data.

1.3 Thesis Organization

This thesis is divided into seven chapters. Chapter 2 gives an introduction to the concepts of genes, genetic interaction, microarray technology, and Woolf and Wang's algorithm. Chapter 3 discusses the ideas behind the proposed improvements to the algorithm. Chapter 4 explains the implementation of the improvements. Chapter 5 illustrates and analyzes the results of the improvements. Finally, Chapter 6 concludes the discussion of the topic and proposes future work on the method.

CHAPTER 2

Genes, Microarray Technology, and Fuzzy Modeling

This chapter discusses the underlying theory behind the research. Section 2.1 covers the basics of how genes work and methods of gene regulation. Additional information about gene theory can be found in [12]. Section 2.2 discusses microarray technology and its pertinence to gene research. Finally, Section 2.3 discusses previous work in fuzzy modeling of microarray data.

2.1 Genes and Gene Interaction

2.1.1 Gene Theory

An individual cell is a complex entity. It must ensure its own survival, maintain its structure, respond to outside stimuli such as changes in temperature and concentration of different substances, and perform functions to keep the cell (and the rest of the organism in a multicellular organism) alive. All of a cell's necessary functions are directly carried out through proteins. Proteins act as structural components or enzymes to catalyze the building or dissociation of compounds that allow the cell to carry out its functions.

The construction of these proteins is achieved through a cell's genetic material. Deoxyribonucleic Acid (DNA), encased within the cell nucleus, is the mechanism by which protein information is stored and passed on to new cells. DNA consists of a chain of compounds called nucleotides that consist of a sugar-phosphate backbone and one of four bases: adenine (A), thymine (T), guanine (G) , and cytosine (C). Solitary strands of DNA are connected through bonds in the sugar-phosphate backbone of each nucleic acid. A pair of strands can connect through each nucleotide binding to its complement base (adenine can only bind to thymine

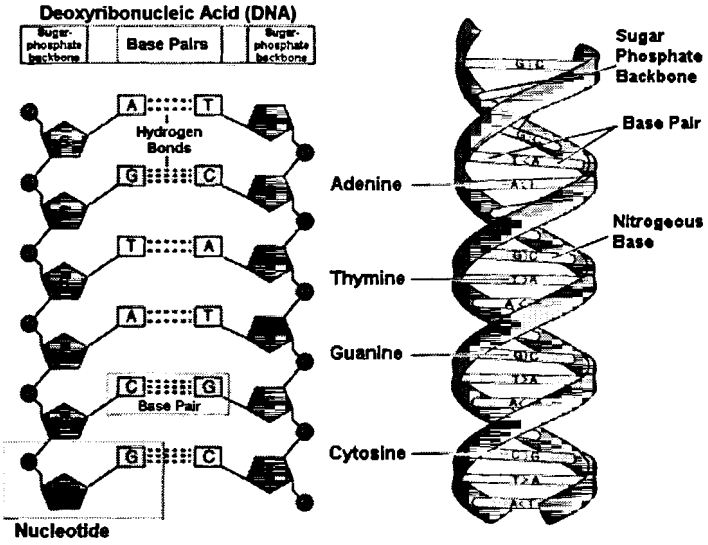


Figure 2.1: Structure of nucleotides and their binding (left) and DNA structure (right) [13]

and vice versa, guanine can only bind to cytosine and vice versa) and coils to form a double helix. The structure of nucleotides and their binding, as well as the three-dimensional structure of DNA can be found in Figure 2.1.

The segment of DNA that contains the code for a particular protein is known as a *gene*. The code itself can be broken down into sequences of 3 nucleotides known as *codons*, each representing an amino acid or a control code (start or stop). From a 1-dimensional perspective, proteins consist of chains (known as *polypeptide chains*) of 20 different types of amino acids. These 20 acids, as well as codons that indicate the beginning and end of a gene, will require a minimum of 22 codons. Since there are four bases in DNA, there are $4^3 = 64$ different codons available. Obviously, there are far more available codons than there are amino acids to code for, so most amino acids have multiple codons, reducing the probability that an improperly reproduced base will result in a different amino acid and thus a different protein.

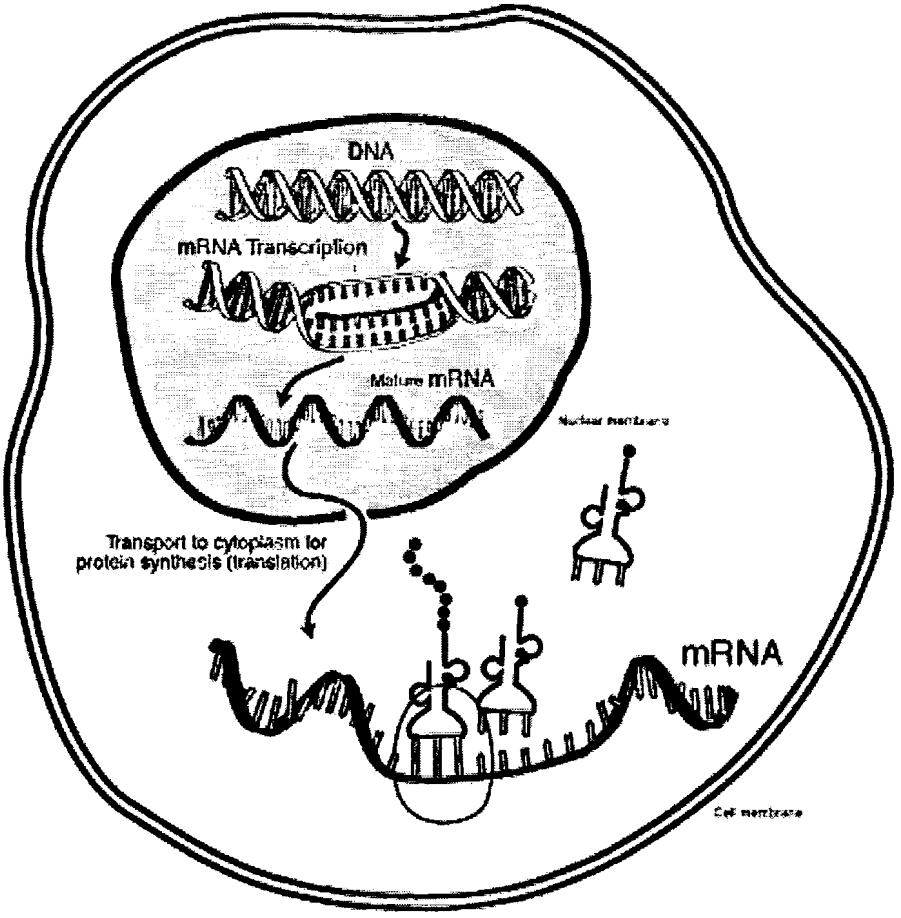


Figure 2.2: Overview of transcription and translation [14]

DNA remains in the nucleus of a cell and cannot synthesize proteins on its own. Ribonucleic acid (RNA) is used as a medium to transmit information from the nucleus to the rest of the cell, where the information can be used to construct proteins. RNA also consists of nucleic acids, but of a slightly different structure than DNA; thymine (T) is replaced with uracil (U) as a base and the sugar in the sugar-phosphate background is of a different type. The process by which proteins are created from the instructions provided by DNA comes in two steps, *transcription* and *translation*. A graphical overview of the processes of transcription and translation can be found in Figure 2.2.

Transcription is the process by which RNA molecules are created for the construction of proteins in the cell. Three types of RNA are created by transcription: messenger RNA (mRNA), which carries the code for a particular protein; transfer RNA (tRNA), which carries a particular amino acid for protein synthesis; and ribosomal RNA (rRNA), which form parts of ribosomes (enzyme complexes that use mRNA to assemble proteins). Transcription of mRNA begins with the enzyme RNA polymerase binding to a region on the DNA strand that indicates the beginning of a gene, known as the *promoter*. Promoters are areas of DNA that allow binding of *transcription factors* (i.e., proteins that bind to DNA, altering its structure and allowing transcription to start at that site.) The process of binding RNA polymerase to a promoter is known as *initiation*. In the next step, known as *elongation*, the mRNA chain is constructed through the binding of RNA bases (adenine, uracil, guanine, and cytosine) to the exposed DNA chain, with RNA polymerase catalyzing the reaction. *Termination* takes place when the RNA polymerase either reaches a Stop codon or a *termination factor* (proteins that bind to DNA to prevent transcription beyond a point.) The mRNA chain may be processed further before being ready to be translated into a protein. tRNA and rRNA are also transcribed in the cell in a similar fashion.

Translation takes place outside the nucleus and involves the creation of a polypeptide chain from a mRNA strand. Ribosomes, which consist of a complex of proteins and rRNA, bind to an mRNA strand. tRNA is used as a medium to bring the necessary amino acids to the ribosome. Many different tRNA sequences exist, each with its own 3-dimensional structure that allows a particular amino acid to bind to it. Each also has a site called an anticodon, that contains the complimentary base sequence to the codon for the amino acid it carries. The anticodon binds to a codon in mRNA, allowing for the binding of the amino acid it carries to the increasing polypeptide chain. When a Stop codon is reached

in the mRNA/ribosome complex, translation stops and the polypeptide chain is separated from the ribosome. The sequence of amino acids determines how the polypeptide folds upon itself and thus its final 3-dimensional structure.

For our purposes, the term "gene expression level" will refer to the concentration of a gene's corresponding mRNA in a cell. tRNA and rRNA must be present in significant concentrations in order for any gene transcription and translation to take place. mRNA, however, varies with how much of a particular protein is to be translated. A certain concentration of mRNA in a cell does not imply that there is a corresponding concentration of its corresponding protein but that there will be at some point in the near future depending upon the rate of translation. In general, higher concentrations of a gene's mRNA sequence in a cell will result in higher expression of the protein encoded by that mRNA sequence.

2.1.2 Methods of Gene Interaction

The expression of a gene and its corresponding protein can be altered, or *regulated*, at several points in the processes of transcription or translation:

1. Initiation of transcription can be controlled by transcription factors that bind to the promoter and allow for easier transcription. Other proteins may bind to transcription factors, altering their structure and rendering them unable to bind to the promoter, thus reducing the volume of transcription.

2. A lack of a particular amino acid or tRNAs to carry the acid will reduce the rate of translation of all proteins according to the number of that amino acid used in the protein. Any genes whose expression is affected by those proteins will also be affected.

3. Presence or absence of different compounds can change the structures of proteins that may directly encourage transcription and/or translation. The

presence or absence of these compounds may be caused by reactions of certain enzymes produced by other genes.

4. Some proteins can bind to mRNA in the cell, preventing its translation into a polypeptide chain.

This list is not exhaustive; there may be other types of regulation. However, we can divide the types of regulation into two major categories: *activators*, whose presence allows for the expression of a gene, and *repressors*, whose presence prevents the expression of a gene.

Activators can work through either positive or negative control. An example of a positive control scenario is an activator binding to a promoter site to allow transcription. A negative control scenario is an activator binding to a repressor, altering its structure to prevent repression of transcription or translation. Repressors can also work through positive or negative control in a similar manner. Activators and repressors need not be proteins; they may be other compounds in the cell that can alter the structure of another protein.

Often, a gene can be directly regulated by several other genes acting as activators or repressors. In these cases, several proteins will combine to form a *complex* that interacts with gene expression processes.

2.2 DNA Microarray Technology

DNA Microarrays [1] attempt to analyze the expression of different genes in parallel on any scale up to the entire genome of an organism.

The construction of microarrays begins with the production of complimentary DNA (cDNA) segments that represent each gene. Each segment is the complement to the actual DNA sequence of a gene and differs from the corresponding mRNA sequence only in that thymine in cDNA replaces uracil in mRNA.

Each spot on the microarray is created by inserting copies of a of one gene's cDNA sequence on a glass slide or other substrate by a high speed robotic process that physically binds the sequence to a small spot on the slide. A spot is created for each gene sequence to be used in the microarray. The substrate and the spots of DNA sequences are collectively known as the microarray. Each spot is referred to as a probe.

To measure gene expression for a cell population, mRNA is extracted from the cells and is reverse-transcribed into complimentary DNA (cDNA). This cDNA sequence is identical to the DNA sequence for the gene found in the nucleus and is thus complimentary to the cDNA probes on the microarray chip. The concentration of each sequence is multiplied proportionally through chemical reactions. Chemical dyes (often green and red in microarray experiments) are bound to the sequences to allow for subsequent analysis of concentration. A solution of this dyed cDNA is created and exposed to the microarray. On the microarray, the cDNA sequences bind, or hybridize, to the probes that contain their complimentary sequence. After a proscribed amount of time, the remaining cDNA solution is washed off the chip. What remains are the probes and the cDNA sequences that hybridized with them. The microarray is scanned with a laser set at the wavelength of the dye's color. The florescent intensity of each spot indicates approximately how many copies of the gene are bound to the spot, and thus, a relative perspective of the expression of that gene in the cell. The appearance of a scanned microarray can be found in Figure 2.3

Unfortunately, the florescence alone tells us very little when the gene expression from only one population is used; we cannot directly correlate the florescence of a probe to the copies of a gene on that probe. To alleviate the problem, we can add a second population whose cDNA sequences were treated with a different dye. This second population can be used as a control population; in the case of time

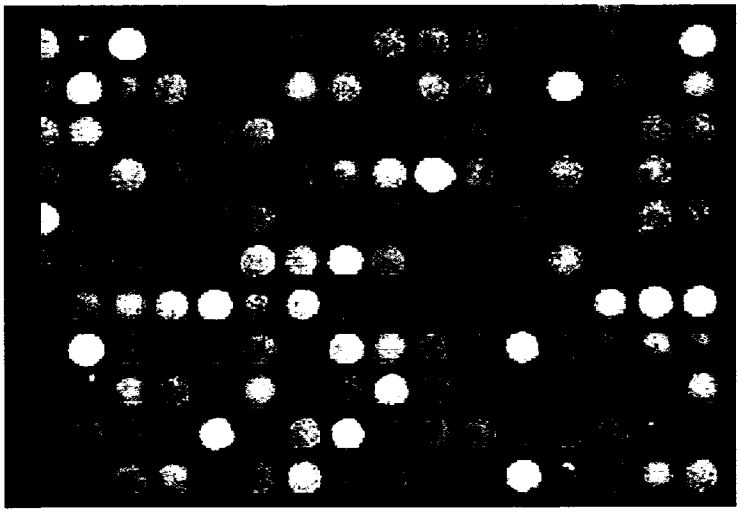


Figure 2.3: A scanned DNA microarray after hybridization

series data, the second (control) population is often the cell population at a fixed point of time while the first population is the same cell population at a later time. The two dyes should have colors of significantly different wavelengths to avoid "crosstalk", i.e., a situation where one dye affects the measured florescence of the other. The relative difference in florescence of the two dyes on a particular spot should tell us how much a gene's expression differs between the two populations. Expression levels can be reported as some form of difference between the two florescences, such as a ratio. Gene expression profiles can be assembled from a series of these differential values at different points in time. The experiments of Spellman *et al* [16] display gene expression timeseries as a listing of the ratios between the experimental and control expression levels for each time point.

The technology is young and still has some problems. First, the florescence signal is unlikely to exactly match the level of expression of each gene. The probe solution used is far from a free solution; the distribution of a certain cDNA sequence through the solution is not even. This problem may be partially alleviated by devoting several spots on the microarray to each gene and averaging the results, but

it cannot guarantee the elimination of the problem. cDNA probes with similar, but not identical, sequences to a particular spot on the microarray may still hybridize to the spot with mixed results, exaggerating the expression of one gene, possibly at the expense of another. Many other issues may also exist. Kerr *et al* [15] identify the sources of signal error:

1. **Array effects** - A time series dataset may be formed over a collection of microarrays, each of which may have differences in cDNA spot concentration, substrate properties, etc.

2. **Dye effects** - A chosen dye might be inherently "brighter" than the other and thus alter the relative florescence between the two populations.

3. **Populations** - This is referred to as "varieties" in [15]. One population may simply have overall higher mRNA concentrations, and thus higher cDNA probe concentrations, due to the nature of the population or a difference in the number of cells used to obtain the mRNA.

4. **Genes** - The importance of a particular differential change for one gene may be higher than that of another gene. Small changes may be important for some genes, but are ignored because they are so small.

Combinations of these four sources of variation can have a significant effect on the relative expression of a gene from these microarray experiments. This variation can be viewed in terms of "noise" in our signal of gene expression for each gene.

2.3 Woolf and Wang's Fuzzy Gene Model Algorithm

As previously mentioned, a fuzzy model of gene interaction would be a generalization of a simple Boolean "on" and "off" model; it realizes that transitionary states exist and attempts to account for them.

Woolf and Wang's algorithm starts by selecting an appropriate subset of genes to analyze. Only genes that meet set minimum expression and differential thresholds are used in the analysis. The differential threshold only accepts genes whose expression changes by a factor of at least 3; that is, the ratio between the gene's lowest and highest expression level should be at least 3 [8]. This ratio exceeds the minimum detectable change (as determined by the estimated noise level) and ensures that the genes used are ones that change significantly over the time series and thus have switched "on" or "off". The minimum expression threshold is set to eliminate genes whose highest expression level is below a certain level. Differential changes are greater for a given absolute change if the overall expression level is low; a difference of 30 between local maxima and minima means a greater differential change when the minimum is 30 than when it is 300. Genes with low expression levels are thus more likely to be distorted by noise and will not serve well in analysis.

Once the subset of genes has been obtained, the data is fuzzified. Each gene is normalized to a scale of 0 to 1, where 1 is the highest expression level and 0 is the lowest (these are the "on" and "off" positions of Boolean data). The normalized data is fuzzified into three fuzzy qualifiers, "Low", "Med", and "High". The membership functions can be seen in Figure 2.4.

All possible triples of genes are applied to a model of gene interaction shown in Figure 2.5. From the model, we get two outputs. The first is the model's output, which is compared to the target gene and scored on basis of the Mean Squared Error between the modeled target output and the actual target output. The second

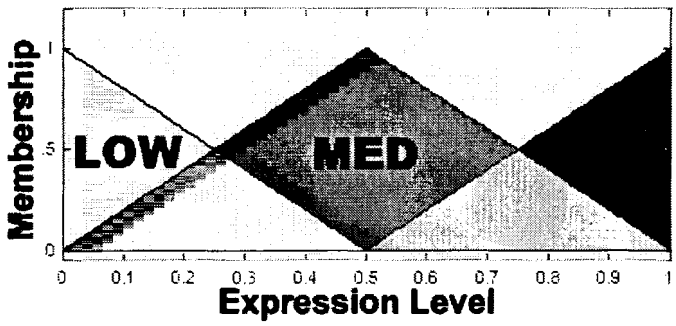


Figure 2.4: Fuzzy membership functions for gene expression data

If activator is \ If repressor is	HIGH	MED	LOW
LOW	target is LOW	target is LOW	target is MED
MED	target is LOW	target is MED	target is HIGH
HIGH	target is MED	target is HIGH	target is HIGH

Figure 2.5: Fuzzy model of gene interaction from [8]

output is a variance; the variance returned is the variance between the total degree to which each rule was fired over the time series. This serves as a confidence value for the model output; lower variances imply that all rules were fired nearly equally and the model output is indicative of the model as a whole. This does not imply that a gene triplet with a high model variance cannot produce a valid output. However, a high variance combination has not been equally handled by all rules in the model and we cannot be sure that the model would fit the triplet in other parts of the model's output space. Thus, the variance serves as a secondary supplement to the error score; it may have some importance when comparing the validity of two gene triplets with nearly identical error scores, but may not be as valid in comparing gene triplets with significantly different error scores. Both the error and the variance are multiplied by 1000 to obtain a score that is easier to read. In both cases, lower scores are better; low scores imply low error or variance.

All possible triplets of genes (with each gene serving in each role in the model) may be examined in this manner. The program records the error and variance of each triplet. To save computation time and memory, error and variance limits can be set. If a triplet has a higher error than the specified limit, it will not be recorded in the results. If a particular activator and repressor combination has a variance above the specified limit, no other triplets with those genes in the activator and repressor positions will be examined.

In [8], Woolf and Wang verified their results through a few methods. First, they attempted to find known gene relationships in the results. As an example in the paper, they were able to find many known relationships to the gene HAP1. Second, they searched for common pairs among the low-error gene triplets. Pairs of genes that appear in many triplets in the same position (e.g., gene A is expressed as an activator and gene C is expressed as a target in many gene triplets), it is likely that the two are related and the third gene in the triplets are relatively irrelevant

to the model. The rationale can apply to either commonly-appearing activator-target or repressor-target pairs. The most frequently appearing pairs were usually biologically related. Finally, they examined the presence of transcription factors in the results. Transcription factors have a direct effect upon transcription of genes and will thus have a profound impact on gene expression. Logically, the lowest-error results should have a disproportionate number of transcription factors; that is, the probability of finding a transcription factor in a low-error gene triplet should be better than the probability of finding a transcription factor in the input dataset.

The appearance of a particular gene triplet in the results does not necessarily mean that the modeled relationship exists between the two genes. Since there are fewer time steps than there are genes, there is no way to develop a unique solution. Due to the stochastic nature of the network, it is unlikely that a unique solution would be right even if one were found. The validity of the results can be further strengthened through the results of different datasets and the union of the results of these disparate datasets to find common links. The model and algorithm make a series of assumptions. First, it is assumed that the time from a gene's transcription to its translation into a protein is negligible; that is, the expression level of a particular gene is directly proportional to the presence of its corresponding protein at any point in time. This is generally not true; reaction times of the system are relatively slow and possibly not constant. If we assume that time from transcription to translation is generally consistent, then we can simply view the gene expression data as the protein expression data with a time shift. Second, the model is *deterministic*, i.e., there can be only one model output for a particular expression level input. However, as discussed in [17], gene interaction is increasingly shown to be a stochastic process: "the number of transcription factors in a cell is often low...the environment in which the gene regulatory interactions occur is far from free solution; and the reaction kinetics is relatively slow." The averaging

of expression levels through using a large number of cells may eliminate some of these effects (by making the overall solution closer to "average" and uniform), but may also distort regulatory networks to some degree [17], hindering our ability to extract important relationships from the genetic network. Care must be taken to ensure that averaging does not distort these networks.

The model is intuitively pleasing; the method is similar to that of a human expert attempting to find relationships through time series data. Fuzzy logic deals with uncertainty and ambiguity; it handles qualitative data and may handle the nonlinear, stochastic nature of the data better than other deterministic models, such as the Boolean model where the gene is either "on" or "off" and nothing in between. It is also expandable to model any known relationships between genes and can be modified to handle time delays or multiple activators and/or repressors.

However, as discussed in the introduction, there are several problems with the model that endanger its viability as an analysis method. First, the algorithm has a high algorithmic complexity; all permutations of three genes must be analyzed, giving the algorithm an $O(N^3)$ complexity. The complexity of the model is directly related to the number of inputs; a model with two activators and two repressors would have an $O(N^5)$ complexity. With each additional input, the run time of the algorithm increases by orders of magnitude. A model with two co-activators has been shown (in our experiments) to take more than 200 times longer than a model with only one activator. Adding another input would likely increase the run time by a similar factor, making the analysis of complex relationships nearly impossible without extremely powerful computers. Second, the output space of the model is highly irregular and thus vulnerable to large changes in output for a small change in input. Since the error of microarray data can be up to 30% or more, it is likely that the output of the model can be highly inaccurate.

CHAPTER 3

Methods to Improve the Fuzzy Gene Modelling Algorithm

This chapter introduces the concepts behind our proposed improvements to the fuzzy gene modelling algorithm outlined in Chapter 1. Section 3.1 discusses how using clusters to approximate groups of gene expression profiles can be used for preprocessing to save run time. Section 3.2 discusses the potential problems of using the algorithm on expression data with high levels of noise and some potential alterations to the model to improve the response. Section 3.3 proposes a general model that can be used to accommodate any number of genes into the model.

3.1 Clustering to Improve Run Time

We can attempt to use gene clusters as metadata for the gene dataset. If a particular combination of clusters does not fit the model well, 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.

3.1.1 Mathematical Formalization

Each gene can be represented as a vector of timeseries data. Suppose $\mathbf{X}$ is an input matrix containing a number of gene vectors $\mathbf{x}_1 ... \mathbf{x}_g$ where g is the number of inputs in the model. Suppose that $\mathbf{y}$ is the output of the model $\mathbf{y} = \mathbf{f}(\mathbf{X})$, i.e., the ideal expression profile of the target gene. If $\mathbf{z}$ is a vector representing the actual expression level of the target gene, the MSE of the model is:

$$MSE(\mathbf{X}, \mathbf{z}) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}_i - \mathbf{z}_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{f}(\mathbf{X}_i) - \mathbf{z}_i)^2 \tag{3.1}$$

Where i is the index into the vector and N is the vector length (i.e., number of points in the time series). Now assume that $\mathbf{X_m}$ and $\mathbf{z_m}$ are meta-data for the input vectors $\mathbf{X}$ and $\mathbf{z}$; that is, $\mathbf{X_m}$ and $\mathbf{z_m}$ contain general information about the expression level that could be provided by using cluster centers of the clusters closest to $\mathbf{X}$ and $\mathbf{z}$. We can now define $\delta\mathbf{X}$, $\delta\mathbf{y}$, and $\delta\mathbf{z}$, where

$$\delta\mathbf{X} = \mathbf{X} - \mathbf{X_m}$$

$$\delta\mathbf{y} = \mathbf{f}(\mathbf{X}) - \mathbf{f}(\mathbf{X_m})$$

$$\delta\mathbf{z} = \mathbf{z} - \mathbf{z_m}$$

Therefore,

$$\mathbf{X_m} = \mathbf{X} - \delta\mathbf{X}$$

$$\mathbf{f}(\mathbf{X_m}) = \mathbf{f}(\mathbf{X}) - \delta\mathbf{y}$$

$$\mathbf{z_m} = \mathbf{z} - \delta\mathbf{z}$$

From these values, we can establish the difference in MSE between a cluster center and its corresponding genes, δMSE:

$$\delta MSE = MSE(\mathbf{X}, \mathbf{z}) - MSE(\mathbf{X_m}, \mathbf{z_m})$$

$$\delta MSE = \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{f}(\mathbf{X}_i) - \mathbf{z}_i)^2 - (\mathbf{f}(\mathbf{X_m}_i) - \mathbf{z_m}_i)^2]$$

$$\delta MSE = \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{f}(\mathbf{X}_i) - \mathbf{z}_i)^2 - ((\mathbf{f}(\mathbf{X}_i) - \delta\mathbf{y}_i) - (\mathbf{z}_i - \delta\mathbf{z}_i))^2] \qquad (3.2)$$

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

$$\delta\mathbf{X} = \mathbf{X} - \mathbf{X_m} \rightarrow 0$$

$$\delta\mathbf{z} = \mathbf{z} - \mathbf{z_m} \rightarrow 0$$

If we assume that $\delta\mathbf{X}$ is small around most input values $\mathbf{X_0}$ and that the gradient of the output space $\mathbf{y}$ is relatively small (which is the case for most $\mathbf{X_0}$, as we will see in Chapter 5),

$$\delta\mathbf{y} = f(\mathbf{X}) - \mathbf{f}(\mathbf{X_m}) \rightarrow 0$$

If we substitute these values into Equation 3.2:

$$\delta MSE = \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{f}(\mathbf{X}_i) - \mathbf{z}_i)^2 - ((\mathbf{f}(\mathbf{X}_i) - \delta\mathbf{y}_i) - (\mathbf{z}_i - \delta\mathbf{z}_i))^2]$$

$$\Rightarrow ((\mathbf{f}(\mathbf{X}_i) - \mathbf{z}_i)^2 - (\mathbf{f}(\mathbf{X}_i) - 0) - (\mathbf{z}_i - 0))^2 \rightarrow 0 \qquad (3.3)$$

Therefore, assuming that we can cluster the data so that most of the genes' expression profiles are relatively close to the 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, 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 center do not fit the model well, thus saving time by not analyzing those combinations.

The results of a version of the algorithm with clustering will always be a subset of the original results; that is, the results will either be identical or missing some gene combinations, but there will be no new combinations of genes in the new results. The method would not directly affect the fuzzy model, so the output of the model for any combination would not change and no new low-error combinations will be introduced. The analysis is not likely to be perfect; due to the output space of the model and the inability of clustering to completely capture the expression profile of a group of genes, it may be possible for a gene combination to fit the model well while its corresponding cluster centers do not. However, if we choose proper selection criteria for which cluster combinations will be searched, we can reduce the likelihood of a low-error gene combination being neglected.

There are a few approaches one could take in determining which cluster combinations are acceptable. One could propose to only analyze a certain percentage of the combinations ranked by MSE or to set an maximum error threshold for cluster combinations to be analyzed. In either method, setting the limits too strictly (i.e., a low percentage of cluster combinations are analyzed or the error threshold is set too low) will result in many valid gene combinations being ignored, while setting the limits too freely (i.e., a high percentage of cluster combinations are analyzed or the error threshold is set too high) will save little time as few invalid gene combinations are ignored. In general, it is favorable to err on the side of caution and set an easily passable limit to obtain a high percentage of at the expense of extra run time.

3.1.2 Clustering Methods

Clustering has already been used on microarray data to find genes with similar expression profiles [2], [3]. The rationale for clustering expression data is that genes with similar expression profiles over several different datasets are

likely to have similar functions. Thus, one can make an educated guess about the function of unknown genes.

Hierarchical clustering [2] has been used to attempt to cluster genes into a hierarchical tree. Upon each iteration of the algorithm, all combination of genes and cluster centers are analyzed under a similarity measure. The two most similar items (two genes, two cluster centers, or one of each) are combined into a cluster center, which replaces the two elements. The process continues until there is only one cluster center. The similarity of two genes can be measured through their distance from each other in the tree. The distance measure used is Euclidean, but other measurement methods can be used. While it has been shown to be an effective tool for visualization of gene expression similarities, it is not intuitive for selecting a set of clusters that is representative of the dataset.

A Self-Organizing Map (SOM) [18] is a clustering method similar to a k-means algorithm, but has a degree of self-regulation through connected networks of centers. Centers can be connected in either a 1 or 2-dimensional network. The SOM training algorithm is similar to that of the k-means algorithm, except whenever a center's value is updated, nearby centers are also updated to a degree proportional to its distance from the updated center in the network topology. The resulting set of clusters is more organized than that of a k-means clustering and is generally more representative of the input space; since the map updates all centers in a particular neighborhood, the cluster centers are more representative of the density of data in different parts of the input space. SOMs have been used to analyze gene expression data of yeast [3] and have been shown to find valid functional groups.

3.2 Changing Fuzzy Methodology to Improve Robustness

There are many factors to be considered when establishing a fuzzy model of gene interaction. The fuzzy rule base, membership functions, methods of fuzzy conjunction and aggregation, and defuzzification can all be changed to accommodate knowledge of gene interaction. The choice of fuzzy model and its mathematical implementation will affect the model's validity and sensitivity to noise. Woolf and Wang's methods of conjunction, aggregation, and defuzzification produce valid results, but are highly susceptible to noise.

Let us reexamine the equations of Section 3.1.1. Suppose that $\mathbf{X_m}$ and $\mathbf{z_m}$ now represent noise-distorted versions of $\mathbf{X}$ and $\mathbf{z}$, respectively. Equation 3.2 states:

$$\delta MSE = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{f}(\mathbf{X}_i) - \mathbf{z}_i)^2 - ((\mathbf{f}(\mathbf{X}_i) - \delta\mathbf{y}_i) - (\mathbf{z}_i - \delta\mathbf{z}_i))^2$$

The difference between the normal and noise-distorted versions of $\mathbf{X}$ and $\mathbf{z}$ can again be expressed as $\delta\mathbf{X}$ and $\delta\mathbf{z}$:

$$\delta\mathbf{X} = \mathbf{X} - \mathbf{X_m}$$

$$\delta\mathbf{z} = \mathbf{z} - \mathbf{z_m}$$

If the error is high, we can no longer assume that either value is near 0. Thus, Equation 3.3 no longer holds and the MSE will be distinctly different if the input data is distorted by noise. We can attempt to minimize the effect by keeping $\mathbf{y}$ continuous and $\frac{\partial \mathbf{y}}{\partial \mathbf{X}}$ small around $\mathbf{X}_0$ so that

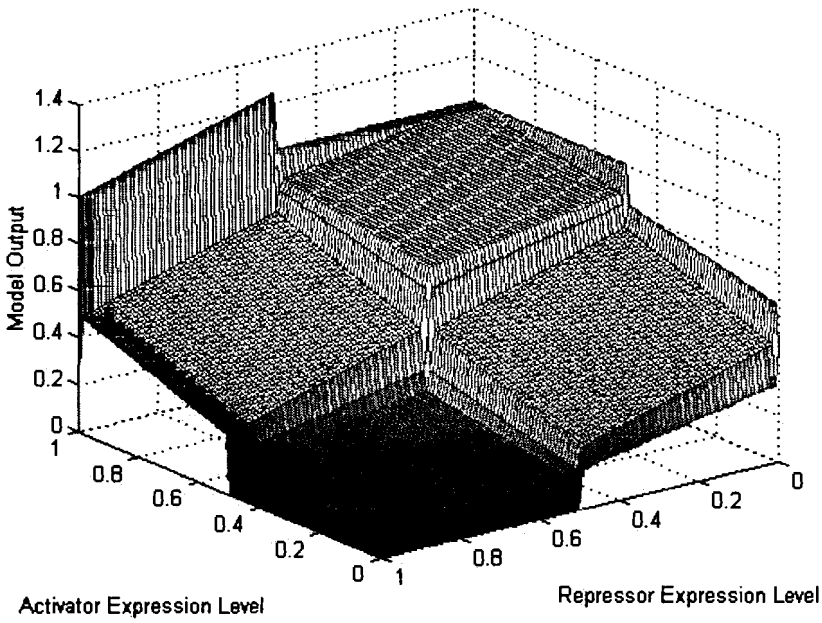


Figure 3.1: Output space of Woolf and Wang's model

$$\frac{\partial \mathbf{y}}{\partial \mathbf{X}} \to 0$$

and

$$\delta \mathbf{y} = f(\mathbf{X}) - \mathbf{f}(\mathbf{X_m}) = \frac{\partial \mathbf{y}}{\partial \mathbf{X}} * \delta \mathbf{X} \to 0$$

However, we must not alter the model so much as to lose its ability to accurately identify activator and/or repressor relationships.

Woolf and Wang's fuzzy model does not fulfill the conditions of a continuous $\mathbf{y}$ or a low $\frac{\partial \mathbf{y}}{\partial \mathbf{X}}$ around many values of $\mathbf{X_0}$. Figure 3.1 represents the output of the model based upon the normalized expression levels of the activator and repressors. We can see that there are several discontinuities in the model and that $\frac{\partial \mathbf{y}}{\partial \mathbf{X}}$ is small only in certain locations of the model output space.

Woolf and Wang's methods of fuzzy conjunction, rule aggregation, and defuzzification are a hybrid of known models; conjunction of membership values is done through a *sum* function, aggregation is achieved by averaging the output of the rules, and defuzzification is done through a modified centroid function.

Different fuzzy model types have been proposed with different methods of conjunction, aggregation, and defuzzification. We will examine two of the model types:

- *Mamdani Model* [19] - 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 set method of 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.

- *Standard Additive Model (SAM)* [20] - Kosko's Standard Additive Model 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.

3.3 Developing More Complex Models

3.3.1 Background

Although application of a simple activator/repressor/target model is helpful, many gene relationships are more complex. For example, the HAT coactivation complex in yeast consists of 11 proteins and thus 11 activators. While it is likely that many pairings of those 11 proteins with the genes they activate would appear in the simple model, extracting the relationship may not be possible as we are only taking one gene into account at a time, rendering the simple model insufficient when examining more complex relationships. A general model must be developed that can accept an arbitrary number of genes as activators and/or repressors.

We propose a generalized version of the model based upon the idea of limiting reactants. All proteins in a complex must be present to form the complex. If one or more of the genes are not expressed highly, the proteins they encode will not be expressed highly, which in turn results in low expression of the complex because certain component proteins are missing. Therefore, if not all of the activators or repressors necessary to activate or repress the target gene of the complex are not highly expressed, it is not likely that the complexes will have a significant effect on the expression of the target gene.

3.3.2 General Model

Again, we will suppose that $\mathbf{X}$ is the input of the model and $\mathbf{y}$ is the model's output. In the simple model, we can divide $\mathbf{X}$ into two vectors, $\mathbf{x_a}$ and $\mathbf{x_r}$, which represent the activator and repressor expression profiles.

We can generalize $\mathbf{x_a}$ and $\mathbf{x_r}$ to $\mathbf{X_a}$ and $\mathbf{X_r}$, which are matrices representing an arbitrary number of vectors of activator or repressor expression profiles. Suppose there exist vectors $\mathbf{x_{ma}}$ and $\mathbf{x_{mr}}$ where

$$\mathbf{x}_{\mathbf{ma}i} = min(\mathbf{x}_{\mathbf{a}1i}, \mathbf{x}_{\mathbf{a}2i}, ..., \mathbf{x}_{\mathbf{a}ji})$$

and

$$\mathbf{x}_{\mathbf{mr}i} = min(\mathbf{x}_{\mathbf{r}1i}, \mathbf{x}_{\mathbf{r}2i}, ..., \mathbf{x}_{\mathbf{r}ki})$$

where j is the number of activators in $\mathbf{X_a}$, k is the number of repressors in $\mathbf{X_r}$, and i=1:N, where N is the number of points in an expression profile. $\mathbf{x_{ma}}$ and $\mathbf{x_{mr}}$ now contain the minimum expression level for the point for all genes in $\mathbf{X_a}$ and $\mathbf{X_r}$. We can assume $\mathbf{x_{ma}}$ and $\mathbf{x_{mr}}$ to be the expression profile of the coactivator/corepressor complex due to the concept of limiting reactants. We can thus use $\mathbf{x_{ma}}$ and $\mathbf{x_{mr}}$ as inputs to the simple model. If the model error is low, we make the same assumption as we did with the simple model: the combination of genes fits the model of gene interaction and is likely to be related in the manner described by the model.

With theoretical foundation established for each section, we can now work on the implementation of the improvements to the model.

CHAPTER 4

Implementation of the Algorithm

This chapter discusses the implementation of the ideas proposed in chapter 3. Section 4.1 explains the experiments performed to analyze the performance of clustering as a preprocessing method. Section 4.2 explains how different model modeling methods were analyzed. Finally, Section 4.3 discusses the analysis and validation of the results of the general model and the effects of clustering in the general model.

Woolf and Wang's algorithm was written in ANSI C and has been run on Unix and Windows machines. Our revisions to the algorithm expand its functionality and perform a series of optimizations for the dataset, including eliminating gene combinations that have a high variance in the model before they are analyzed, changing the order in which triplets are analyzed so that the model only needs to be applied once per activator/repressor pair, and other code optimizations.

4.1 Clustering to Improve Run Time

Three public timeseries datasets were obtained from experiments in [9] and [16]. All three datasets were of *saccharomyces cervisae* under different conditions, herein referred to as cdc15, cdc28, and elu experiments. Selection and normalization of genes was performed in an identical manner to Woolf and Wang's experiment [8]. All expression profiles that made the necessary expression cutoffs were filtered to eliminate high-frequency noise and extract the general shape of the expression profile. The filtered data was clustered using GeneCluster, the SOM software developed for [3]. Several runs were performed for each number of clusters used and the results with the lowest variation between cluster nodes and the data were selected. The number of nodes was increased until the decrease in

standard deviation between genes and their corresponding clusters was minimal. The product of clustering was two files: a file of cluster node profiles (File 1) and another file of gene expression profiles and the ID of the most similar cluster node (File 2).

The cluster node file was run through a modified version of the algorithm that views the nodes as "genes" in their own right. All possible combinations were analyzed and scored on basis of error and rule variance as discussed in Chapter 2. The program produces a third file (File 3) that contains each combination of cluster nodes and its corresponding error and variance in the model. We thus have our evaluation of the cluster nodes that should give us some insight as to which combinations of genes will fit the model well.

The unfiltered, normalized data is reintroduced to File 2 and the algorithm is rerun using Files 2 and 3. Before analysis of gene combinations, each cluster combination and its error and variance in the model is stored locally. Analysis of gene combinations then commences. Before analyzing a particular triplet of genes, the nodes of the clusters they belong to are evaluated in one of two manners, depending upon the experiment:

1. 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. 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, the gene triplet is not analyzed and the algorithm proceeds to the next triplet.

For the percentile ranking method, experiments were run with an error cutoff of 2000 (implying 2% MSE) and a variance cutoff of 40000 for all combinations of:

1. All three datasets.
2. Numbers of clusters ranging from 4 to 15
3. Ranking percentile cutoffs of the top 50%, 60%, 67%, and 75% of combinations.

For the error threshold method, experiments were run with gene error cutoffs of 1500, 2000, and 2500 (MSEs of 1.5-2.5%) for all combinations of:

1. All three datasets.
2. All numbers of clusters ranging from 12 to 15
3. Cluster error thresholds at several points between 7000 and 12000 (implying 7-12% MSE)

The reason for a smaller range of clusters in the error threshold experiments is due to the number of experiments required for any number of clusters as well as evidence from the percentile cutoff experiments (which occurred before the error threshold experiments) of marginal returns beyond a certain number of clusters. Several different error cutoffs were run to see its effect on optimal threshold.

Each experiment was timed starting at the beginning of the loop of analyzing triplets and ending at the end of the loop. The number of gene triplets that passed error and variance cutoffs and the time required to run the program were compared against an experiment with the percentile cutoff at 100% (no triplets were ignored) or an error threshold of 100000 (implying an MSE of 100%, meaning no triplets were ignored) The percentage of the full results obtained by experiment as well

as the percentage of the full time required were stored. The data was graphed as the percentage of full combinations and time required as a function of the number of clusters and the percentile cutoff (or error threshold cutoff) selected. The percentile cutoff experiments were represented in two-dimensional graphs with each line representing a different percentile cutoff. The error threshold experiments were represented as a three dimensional graph with the threshold value and number of clusters as the independent variables.

4.2 Changing Fuzzy Methodology to Improve Robustness

Four different fuzzy models were tested for sensitivity to noise:

1. Woolf and Wang's original model (sum conjunction, average aggregation, modified COA defuzzification)
2. Mamdani Model (min conjunction, max aggregation, clipping COA defuzzification)
3. Standard Additive Model (product conjunction, sum aggregation, scaling COA defuzzification)
4. Hybrid Model (product conjunction, max aggregation, scaling COA defuzzification)

The model output surfaces were calculated for each of the models. The gradient was calculated at increments of 0.01 and the mean and standard deviation of the model gradient was calculated from those datapoints. The results of the gradient analysis were compared against each other to compare typical model gradient.

A Monte Carlo simulation was run to find the effect of noise on the model. The output of the original algorithm was used as a basis for gene triplets to use in our analysis. For each gene triplet in the results:

1. The unnormalized data for each gene in the triplet is extracted from the original datafile.

2. Each timepoint in each of the three genes is distorted by a random amount of noise up to a specified noise limit expressed as a percentage of the current expression level.

3. The distorted data is normalized and applied to the model.

4. The new model error and variance are calculated.

Each gene triplet goes through 20000 iterations of the process. The mean and standard deviation of the error and variance for the 20000 iterations are stored along with the original error and variance of the "noise-free" triplet ("Noise-free" is in quotes because the original gene expression data is itself distorted by noise; we only assume that it's not for the sake of experiment)

The Monte Carlo simulation was run for noise limits at increments of 5% from 5% to 35% and on four different models. The distribution of error plots were made for the mean and standard deviations of noise-distorted MSEs versus the original MSEs for each model and noise limit and plotted a regression line on the new mean MSE versus the original MSE. Sensitivity to noise can be found by checking the equation of the regression line; a model is less sensitive to noise if the slope of the regression line is close to 1 and the y-intercept is close to 0. A slope of 1 would imply that on average, all model outputs would be distorted by the same factor, regardless of the original error score. 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 a constant due to error, we can simply raise our error cutoffs to get the same results. Ideally, we would want to find the regression line $y = x$, where x is the original error score, while y is the error score after noise distortion. Such a regression would imply that the average change in MSE is 0. However, a model that yields the ideal regression line may not be sufficient to produce reliable results; it may not be sensitive enough to properly model gene interaction. Some compromise between the ideal and Woolf and Wang's model is desired.

To ensure that reduced noise sensitivity does not affect the validity of the results, we attempted to validate the results for each model in a manner similar to [8]. We checked for the detection of certain relationships mentioned in Woolf and Wang's paper, analyzed the enrichment of transcription factors, and looked at the relationships between the most commonly appearing pairs of known genes.

4.3 Developing More Complex Models

The generalization of the fuzzy model to accommodate any number of activators or inhibitors is relatively simple. The algorithm checks every combination of $A + B + 1$ genes, where A is the number of activators and B is the number of repressors. If there are multiple activators or repressors, the expression profiles are combined using the min operator. The resulting three expression profiles (activator complex, repressor complex, and target) are applied to the model and analyzed in the same manner as with the simple model.

Clustering experiments were performed and analyzed in a manner identical to the simple model for a model with 2 activators and 1 repressor using the *saccharomyces cervisae* data from [9]. The number of experiments was reduced due to the time required and the lack of computational resources. As such, only the following experiments were produced:

1. cdc28 dataset only.

2. Ranking percentile cutoffs of the top 50%, 67%, and 75% of combinations with a number of clusters ranging from 4 to 15.

3. Error cutoffs at several points between 7000 and 12000 (implying 7-12% MSE) with only 15 clusters.

Again, the number of clusters used for the percentile cutoff experiments are more complete because they were completed first; it became obvious that the number of clusters to use should be above a certain level.

Validation on the model was also performed in a similar manner as described above. Enrichment of known coactivators was also performed; known coactivators should be in a large percentage of results relative to the number present in the input dataset.

CHAPTER 5

Results and Analysis

This chapter reveals and analyzes the results of the experiments proposed in Chapter 4. Section 5.1 shows the results of the experiments on clustering as a preprocessing method. Section 5.2 shows the output space, Monte Carlo simulations, and model validations for each of the proposed models. Section 5.3 shows the validation of the general model and the effects of clustering on the general model.

5.1 Clustering to Improve Run Time

Results in this section are obtained using the Mamdani fuzzy model for reasons explained in Section 5.2. As shown in [25], similar results were obtained using Woolf and Wang's original model.

5.1.1 Clustering Observations

Repeated clustering of each dataset with different numbers of clusters and learning rates revealed that the overall variance of gene time series around cluster nodes changes little over different runs. Common results for standard deviations of genes around a cluster node as a function of the number of genes can be found in Figure 5.1.

It is apparent that the gradient of each curve approaches 0 as the number of clusters increases and appears negligible in all of the datasets at about 12 clusters. This observation is fortunate; the amount of memory required to store the ranking is exponentially related to the number of inputs to the model. As the number of inputs increase, as is the case with the general model, the amount of extra

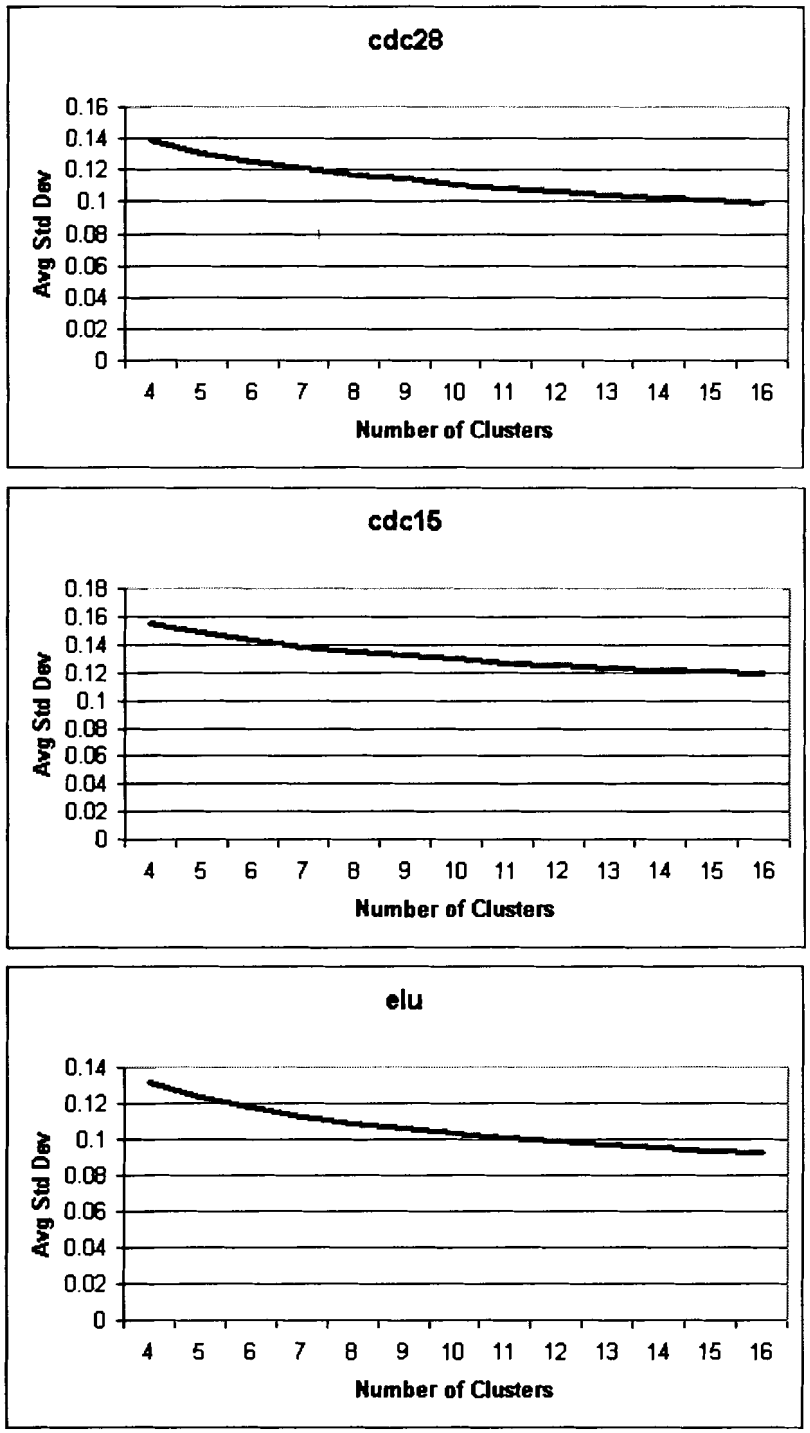


Figure 5.1: Standard deviation of gene timeseries around cluster nodes

space required to store ranking data for each additional cluster node becomes prohibitively large.

5.1.2 Cluster Analysis - Percentile Cutoff

Figures 5.2-5.4 depict the percentage graphs discussed in Section 4.1.

The results were obtained assuming an error cutoff score of 2000 (implying an MSE of 2%) and a variance score of 40000.

The most apparent observation is that the percentage of valid combinations obtained increases as the number of clusters increases (and the standard deviation between a gene and the nearest cluster center decreases.) However, the gains obtained by increasing the number of clusters is in steady decline, which is consistent with the findings that improvements in standard deviation around clusters decrease as the number of clusters increases.

Except for a few of the experiments on the cdc15 dataset, it is obvious that the time taken by the algorithm is relatively constant regardless of the number of clusters used. The time required is primarily dependent upon the percentile cutoff used. The time saved is obviously not identical to the percentile cutoff, but this can be excused as some overhead for checking each gene combination for the cluster cutoff may account for some of the extra time.

Another observation is that the results depend heavily upon the dataset used. For a particular number of clusters and percentile cutoff, the results are far different for the cdc15 dataset than they are for the elu dataset. This fact hinders the use of this method of choosing gene combinations; the only way to find an optimal number of clusters and percentile cutoff is to run the algorithm repeatedly, which will take longer than simply using the original algorithm.

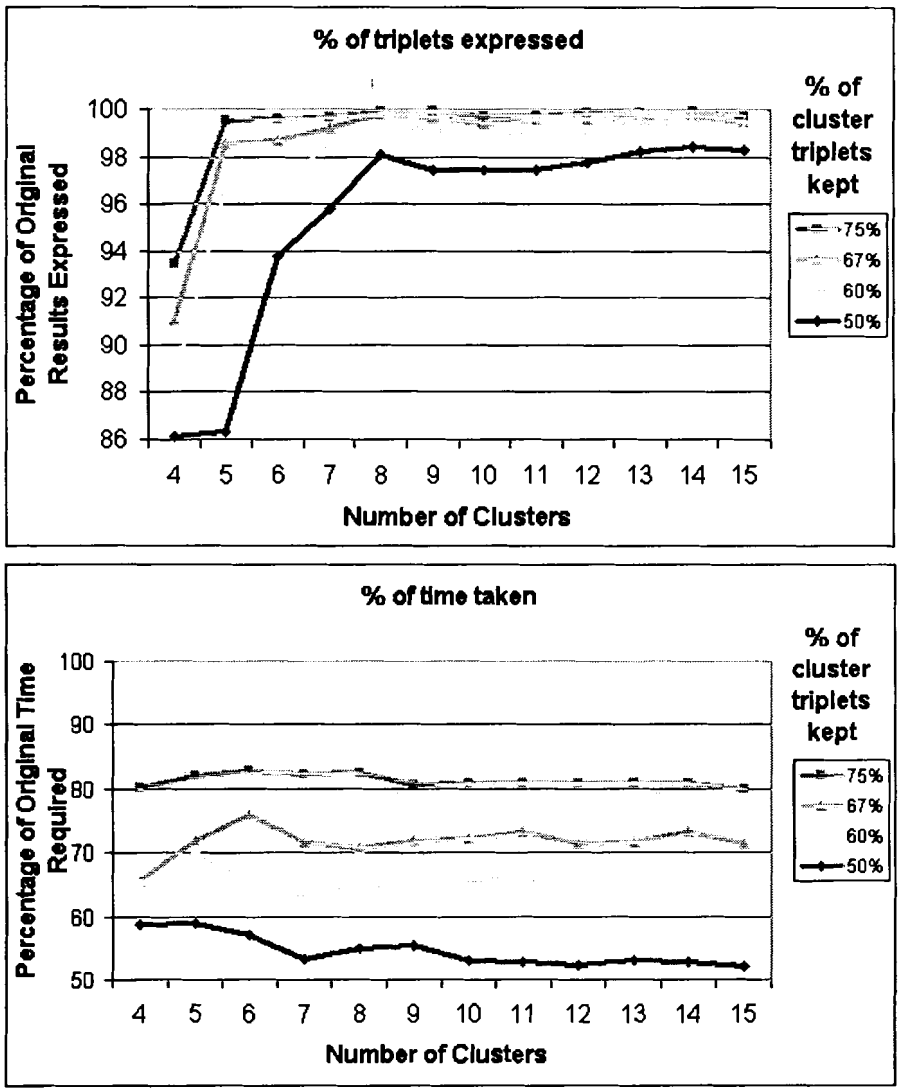


Figure 5.2: Results obtained and time required for clustering method of fuzzy analysis for cdc28 dataset [9] using cluster error percentile cutoffs

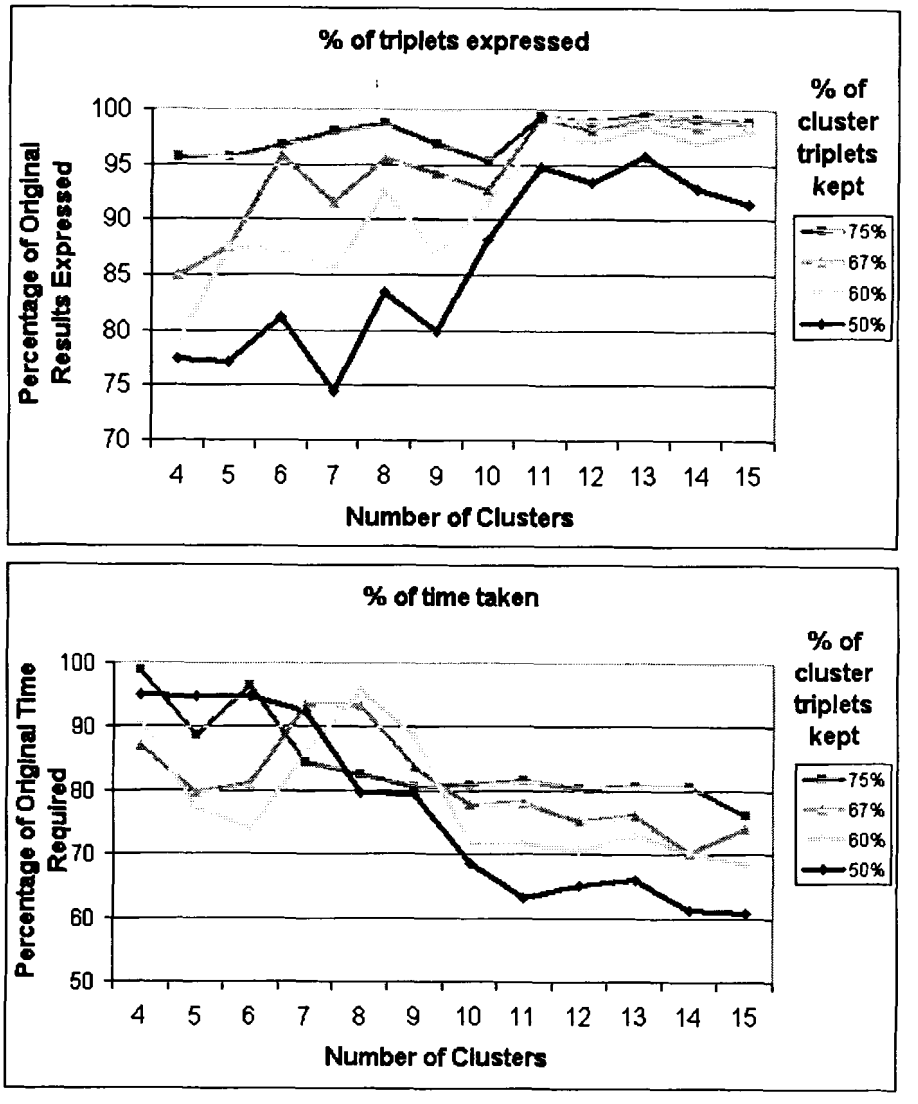


Figure 5.3: Results obtained and time required for clustering method of fuzzy analysis for cdc15 dataset [2] using cluster error percentile cutoffs

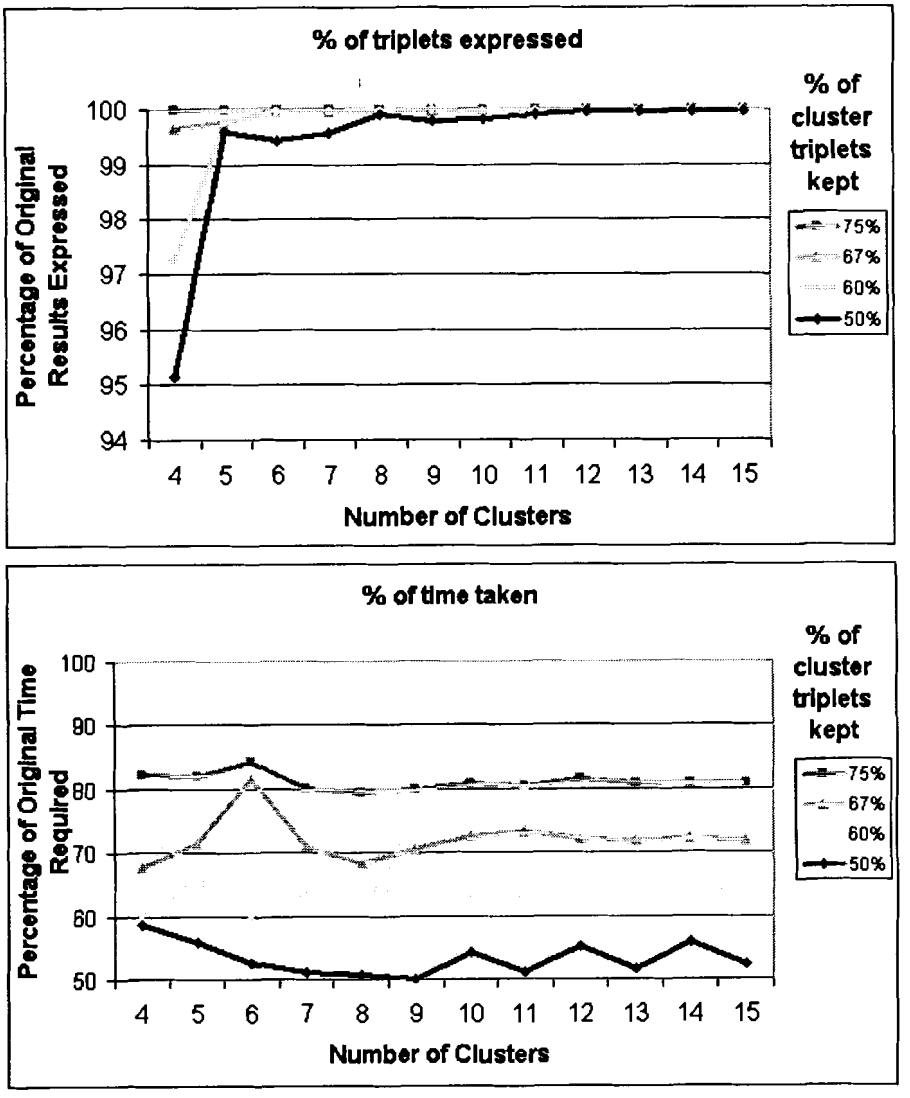


Figure 5.4: Results obtained and time required for clustering method of fuzzy analysis for elu dataset [2] using cluster error percentile cutoffs

5.1.3 Cluster Analysis - Cluster Error Threshold

A disadvantage of selecting percentages of cluster combinations, as seen in the previous section, is that selecting the percentage for optimal time saving and results is completely subjective to the dataset. Thus, a different method needs to be used to reliably obtain most of the results independent of the dataset.

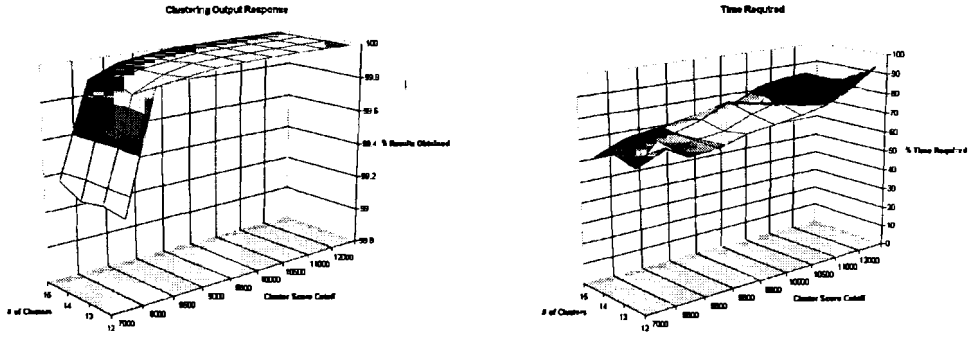
It was observed that, for a particular percentage of the original results obtained, the error score of the worst combination of clusters that was checked was relatively constant. This fact led to the idea of using a cluster error score threshold to select the cluster nodes whose corresponding genes would 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.

Graphs summarizing the percentage of results obtained relative to the original results and the percentage of the original time required can be found in Figures 5.5 - 5.7. The percentage of the original results and time are displayed as a function of the number of clusters used and the cluster error threshold.

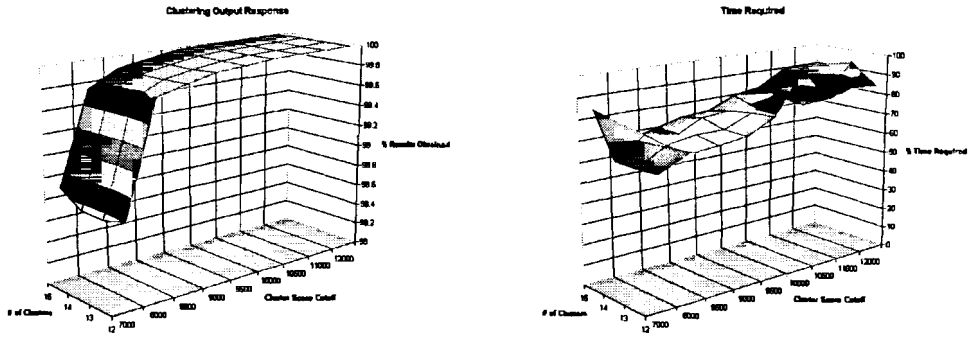
To analyze the accuracy of the algorithm in identifying valid gene combinations, we will define a "99.9%" point, which is the lowest cluster error threshold that returns 99.9% of the original algorithm's results.

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 a threshold of 7500 for a desired cutoff of 1500, 8000 for a desired cutoff of 2000, and 8500 for a

Error score cutoff: 1500



Error score cutoff: 2000



Error score cutoff: 2500

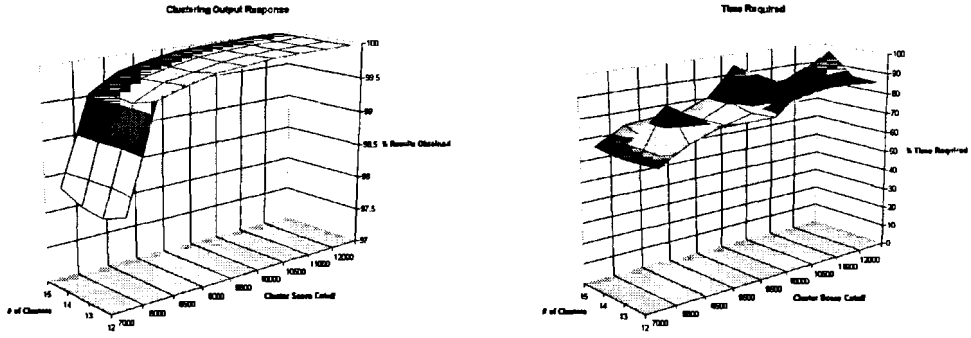
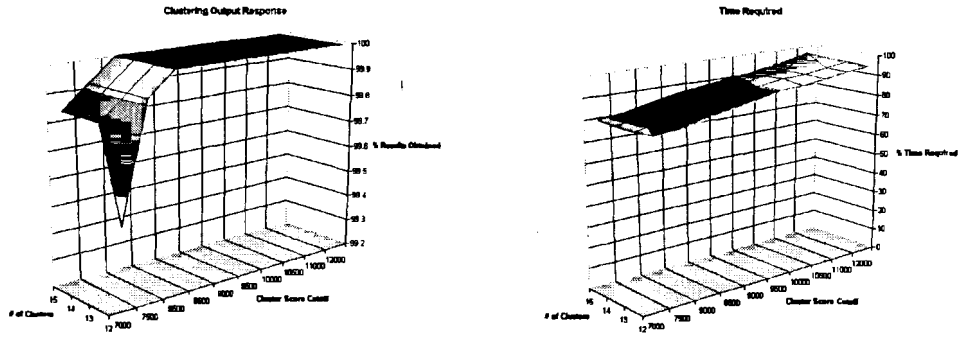
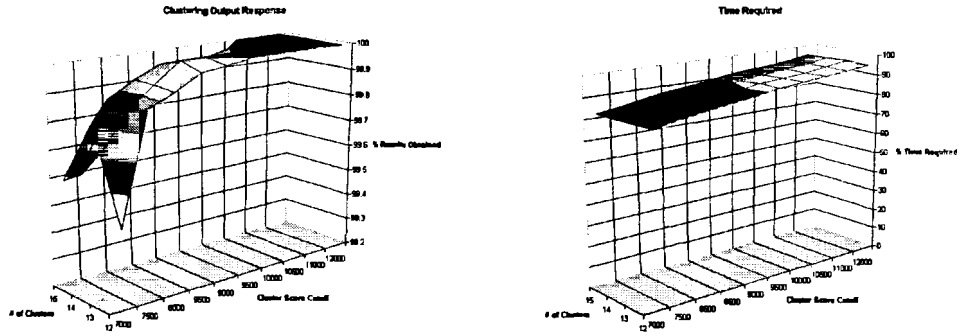


Figure 5.5: Results obtained and time required for clustering method of fuzzy analysis for the cdc28 dataset in [9] using absolute cluster error thresholds

Error score cutoff: 1500



Error score cutoff: 2000



Error score cutoff: 2500

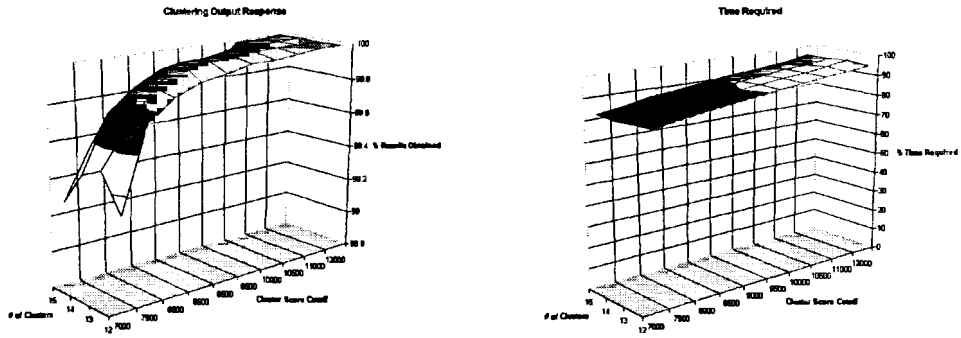
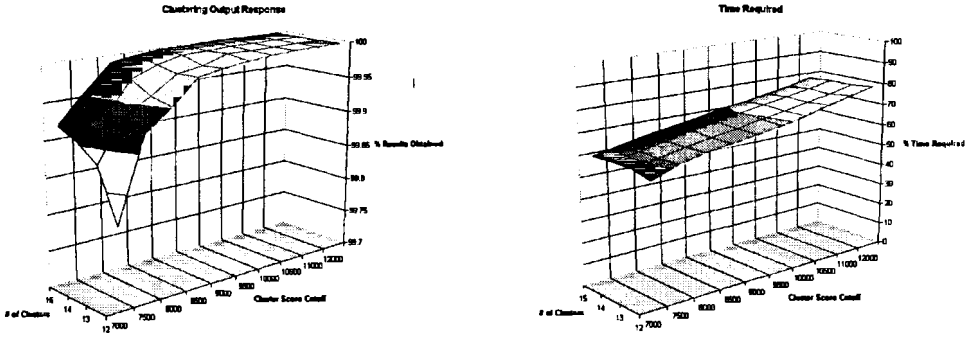
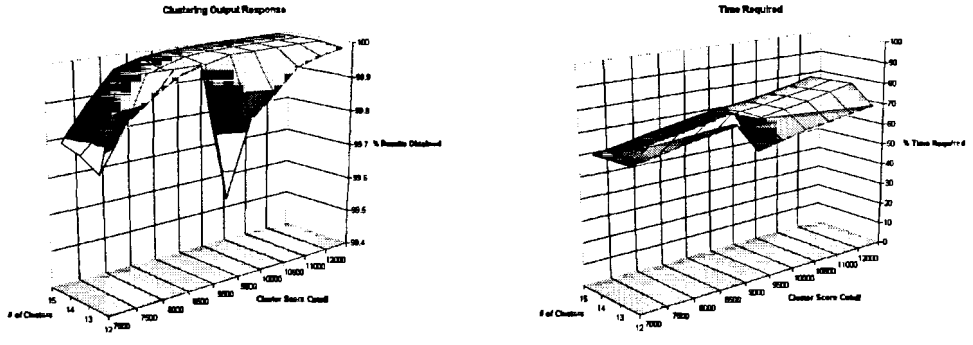


Figure 5.6: Results obtained and time required for clustering method of fuzzy analysis for the cdc15 dataset in [2] using absolute cluster error thresholds

Error score cutoff: 1500



Error score cutoff: 2000



Error score cutoff: 2500

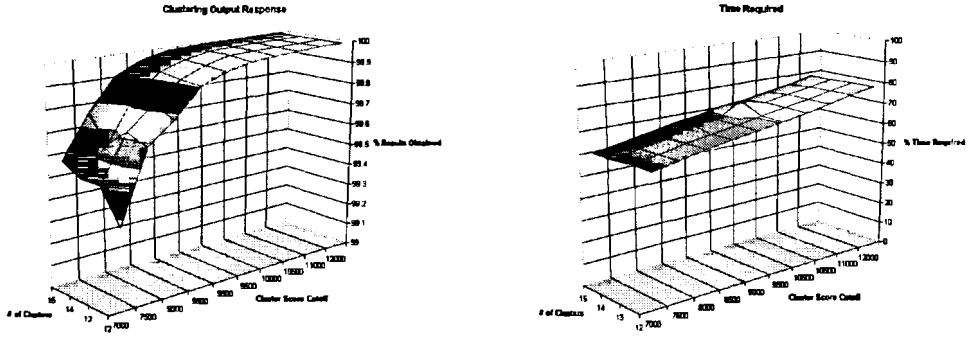


Figure 5.7: Results obtained and time required for clustering method of fuzzy analysis for the elu datasets in [2] using absolute cluster error thresholds

desired cutoff of 2500. For the range of gene combinations we would be most interested in (i.e., those with an MSE of 2.5% or less), we find that the error threshold is linear and independent of factors other than the desired error cutoff.

As made evident by Figures 5.5 - 5.7, 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 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 the complexity of the dataset. However, the inability to forecast the run time is countered by the increased ability to forecast the percentage of the original results produced by the new algorithm.

5.2 Changing Fuzzy Methodology to Improve Robustness

5.2.1 Gradient Analysis

The output space of the four fuzzy modelling methods (Woolf and Wang, Mamdani, SAM, and hybrid) can be found in Figures 5.8 - 5.11.

As can be seen in Figures 5.9 - 5.11, the three alternate models proposed have similar output spaces with only minor variation. Because of centroid defuzzification, none of the alternate models can produce a model output of 1 or 0 . Thus, there will always be some error between a target gene and the model output as one time point in the gene's expression profile will have a value of 1 and another timepoint will have a value of 0.

An analysis of the gradient of the output space of each model can be seen in Table 5.1. The highly irregular response of Woolf and Wang's model is reflected in

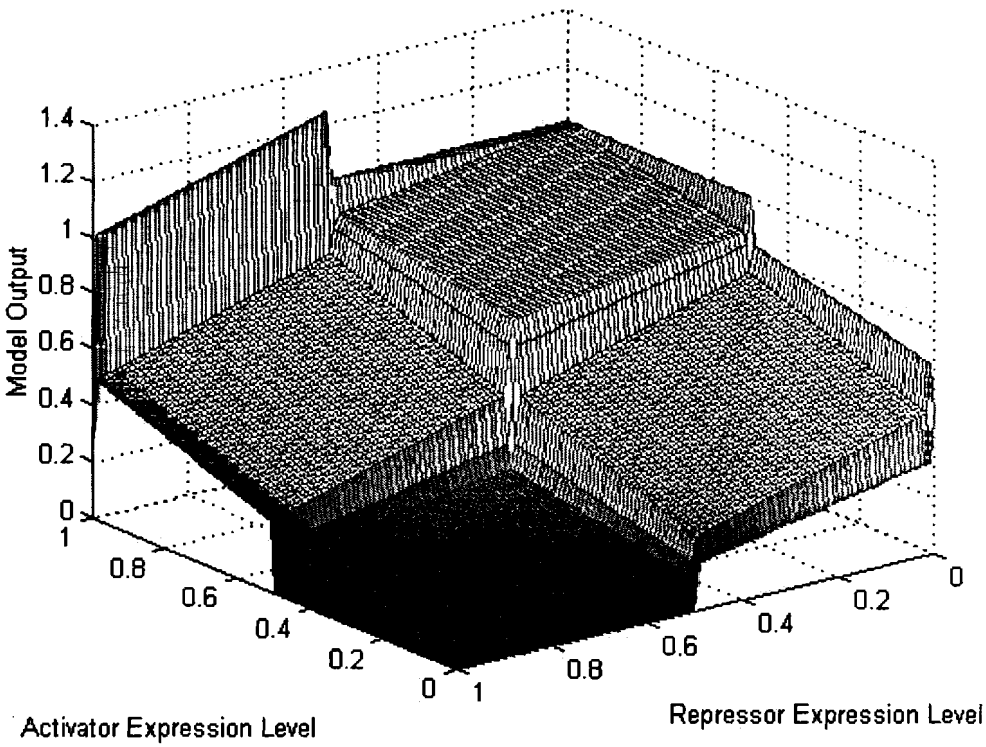


Figure 5.8: Output space of the Woolf & Wang model

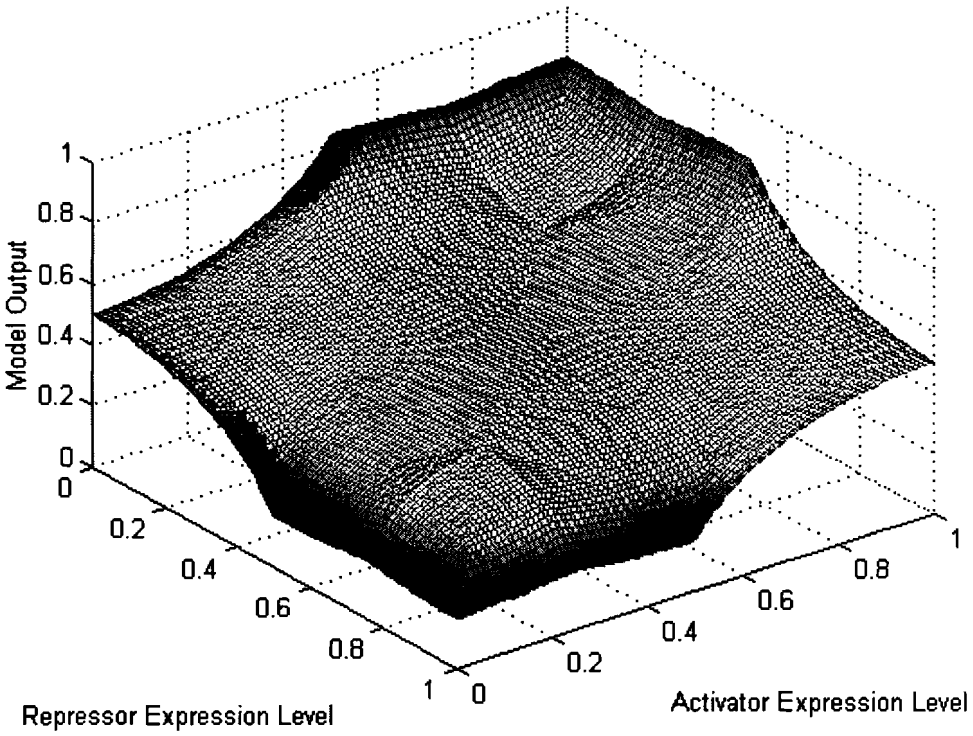


Figure 5.9: Output space of the Mamdani model

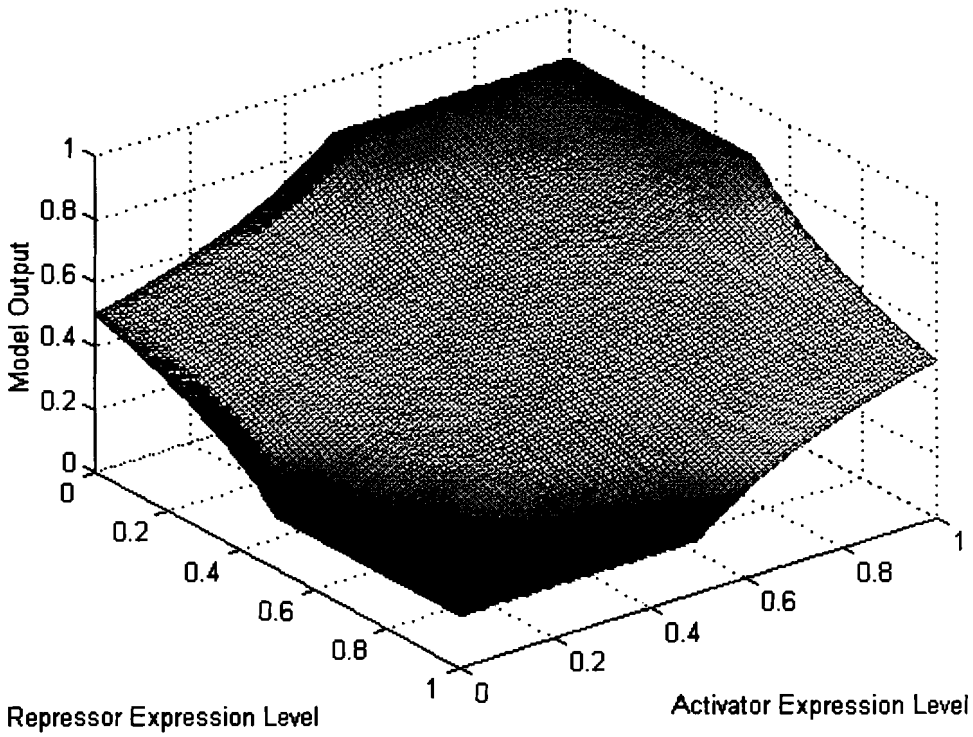


Figure 5.10: Output space ofthe Standard Additive model

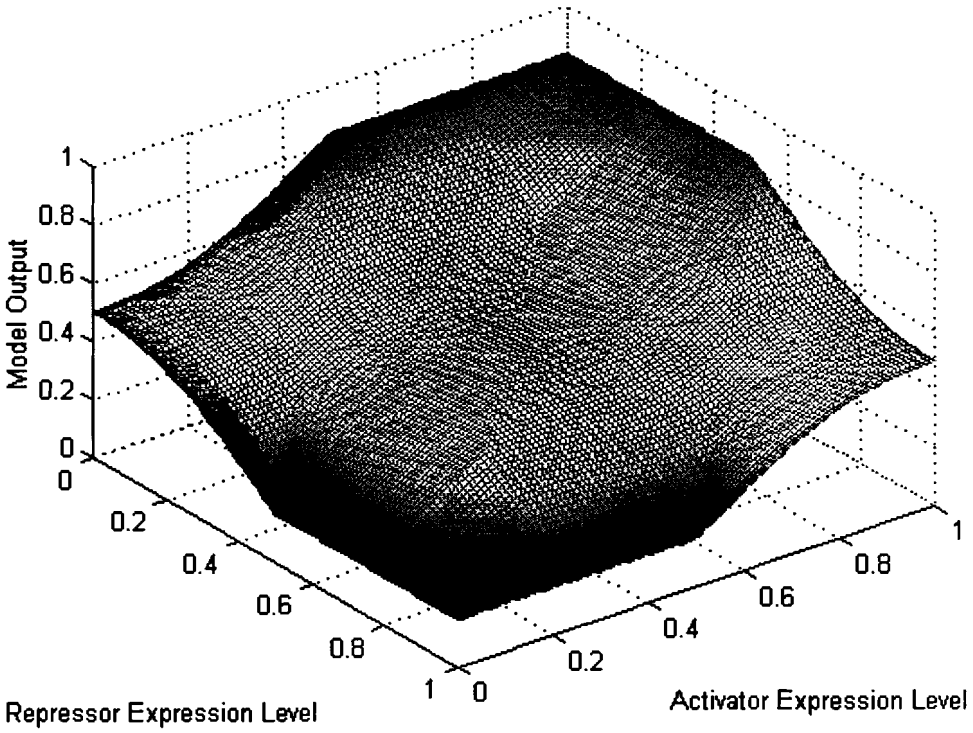


Figure 5.11: Output space of the Hybrid model

Model Type	Mean Gradient	Std Dev of Gradient
Woolf & Wang	10.31	13.68
Mamdani	5.48	4.64
SAM	6.57	0.68
Hybrid	6.02	2.88

Table 5.1: Gradient analysis of the fuzzy models.

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 offers a much lower average gradient and standard deviation. The Standard Additive Model 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.

5.2.2 Monte Carlo Error Simulations

Error simulations for 5% and 30% noise for each of the models can be found below in Figures 5.12 - 5.15. A more complete set of simulation graphs can be found in Appendix A.

From the graphs, it appears that Mamdani model produces regression lines with the slope closest to 1 for all potential noise distortions. This implies that, *on average*, the primary effect of noise on the model is to add a constant error offset to the noise-free error score. The original fit of the inputs (i.e., the noise-free error score) has little or no effect upon the noise-distorted data's fit of the model. If the standard deviation of noise-distorted error scores is also low, as is the case with the Mamdani model, 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 [15], one could, while checking the results of the algorithm for gene combinations with a certain error score cutoff, raise the desired error cutoff

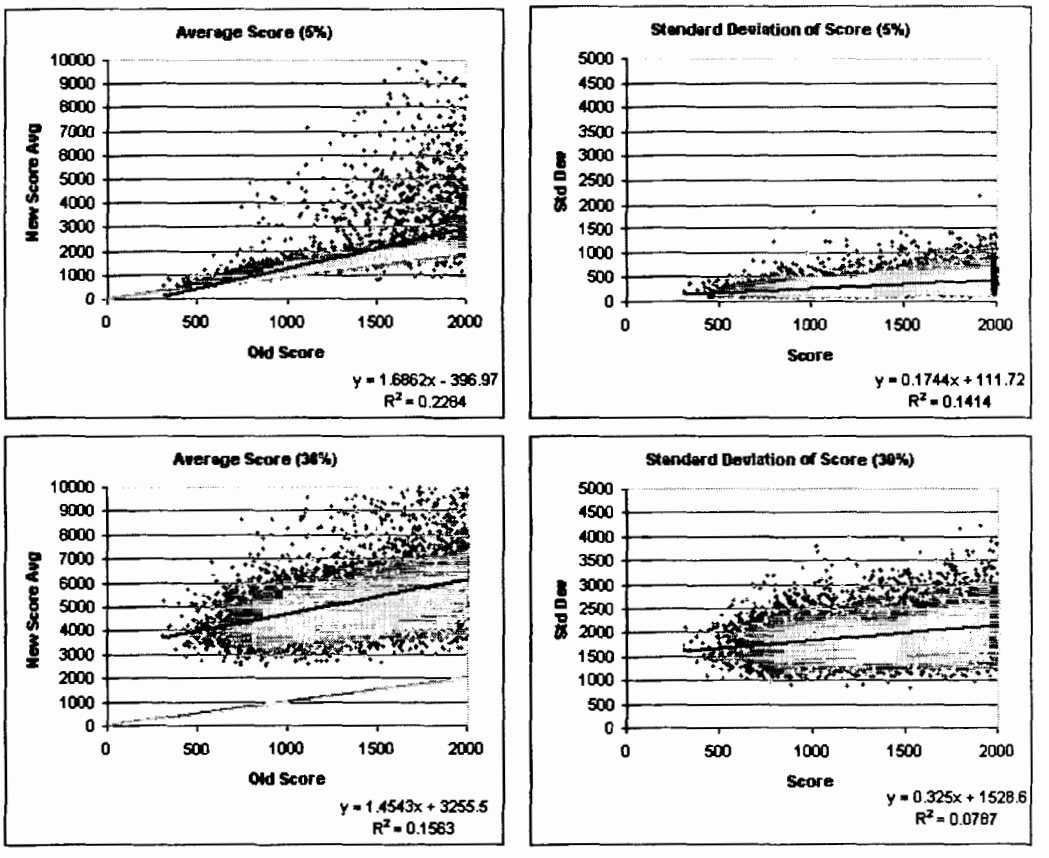


Figure 5.12: Monte Carlo error simulations for the Woolf & Wang model

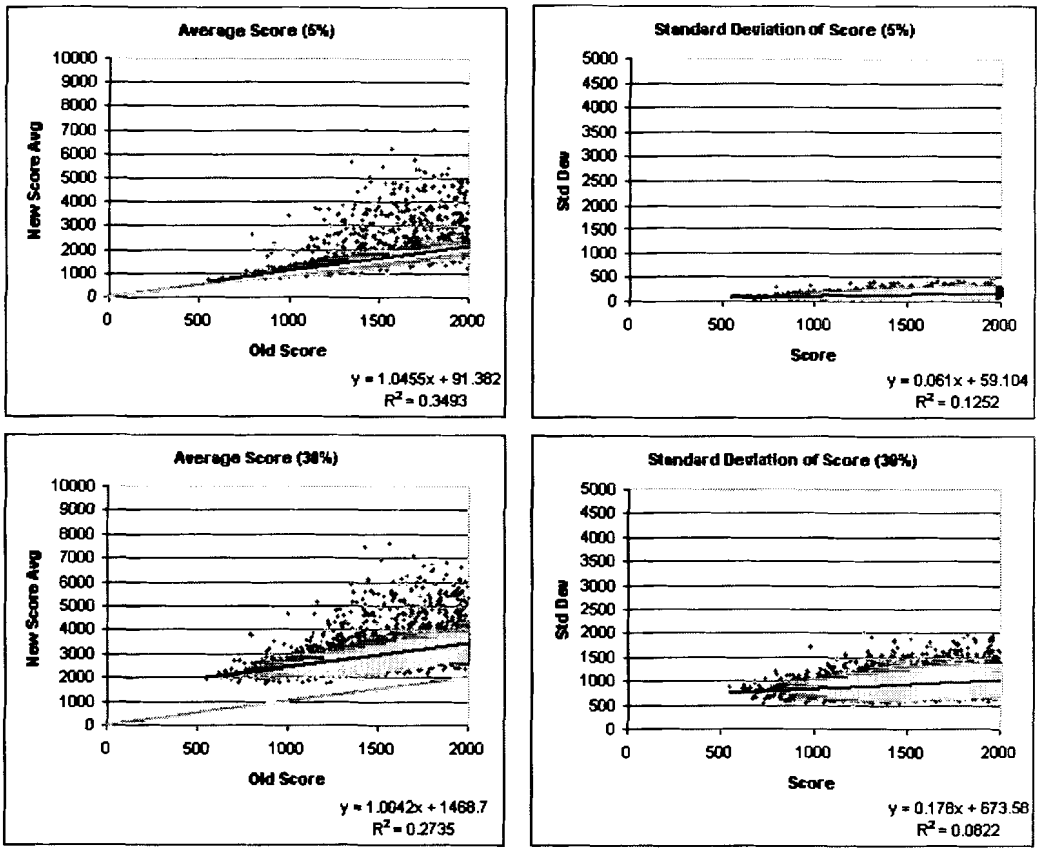


Figure 5.13: Monte Carlo error simulations for the Mamdani model

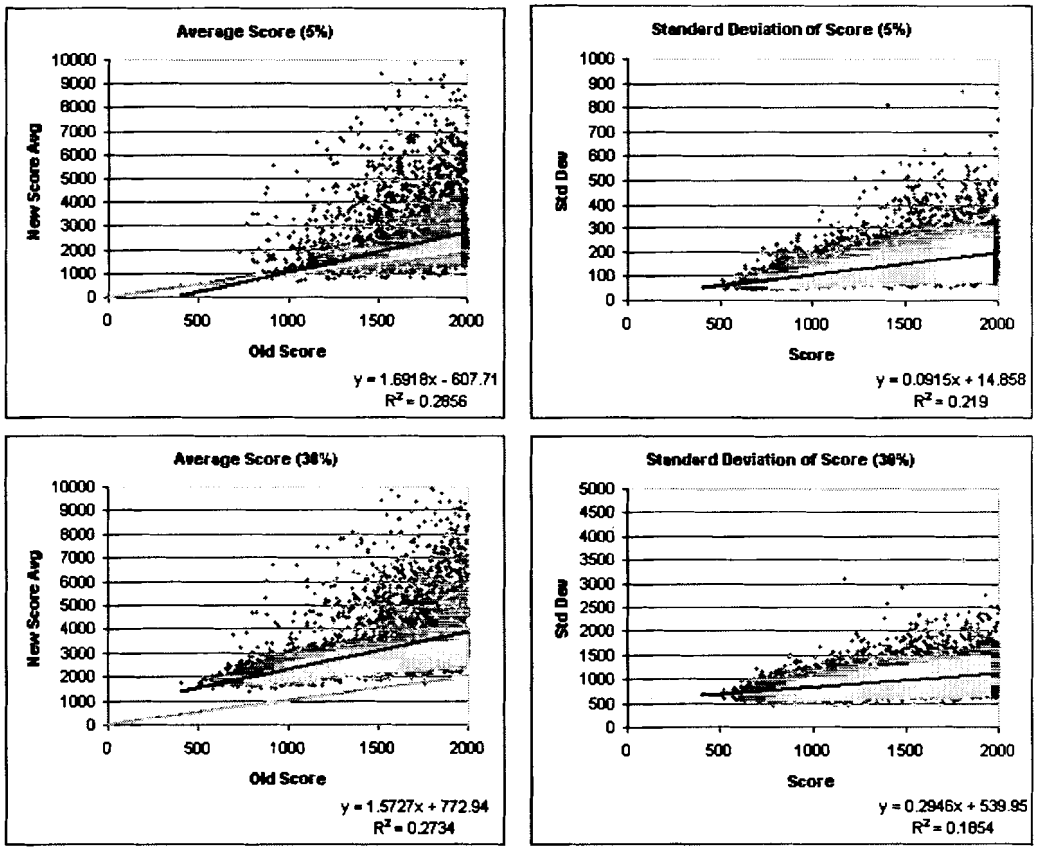


Figure 5.14: Monte Carlo error simulations for the Standard Additive Model

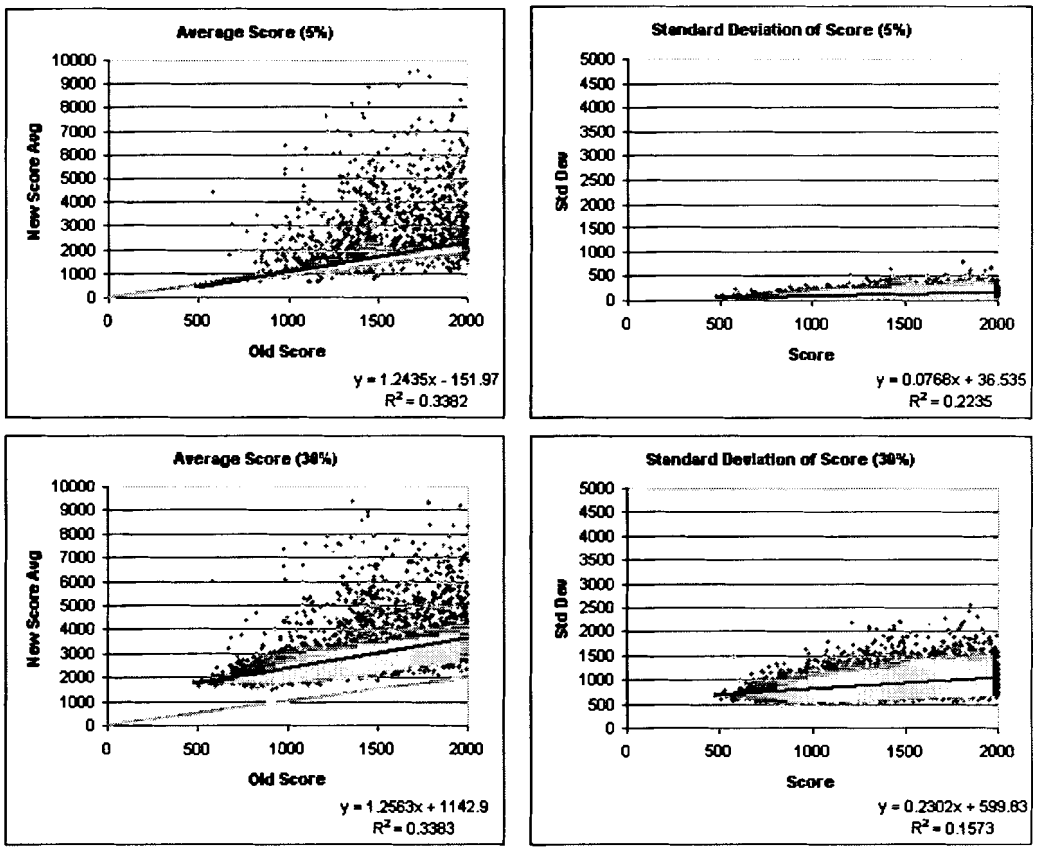


Figure 5.15: Monte Carlo error simulations for the Hybrid model

Model Type	TF % in results	Ratio of Enrichment
Woolf and Wang	8.96%	2.26
Mamdani	10.53%	2.66
SAM	9.43	2.38
Hybrid	10.51	2.65

Table 5.2: Transcription factor enrichment of the fuzzy models.

by the value of the constant error offset to obtain the majority of genes that are likely to fit the model under noise-free conditions.

The other three models (Woolf and Wang, SAM, and Hybrid) have regression line slopes significantly greater than 1, implying that high-error gene combinations will be distorted by a disproportionate amount in the presence of noise. This makes analysis of results less reliable and more difficult. The standard deviation of results around their means are also much higher. In general, the Woolf and Wang model has the highest slopes and standard deviations. The Standard Additive model has slightly lower slopes and standard deviations. The Hybrid model produces results between the SAM and Mamdani models.

5.2.3 Model Validation

Transcription factor enrichment results can be seen in Table 5.2. The percentages are derived from the results of each model with an error score cutoff of 2000 (MSE of 2%) and a variance cutoff of 20000. 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 in the cdc28 dataset).

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

A	B	C	No.	Functions
—	PUS2	LEU4	273	PUS2 alters tRNA-Leu, which inhibits LEU synthesis genes [23]
—	GSC2	LEU4	156	Involved in different metabolisms. Only one should be active.
—	PUS2	ARO3	145	PUS2 alters tRNA-Tyr, which inhibits translation of ARO3
—	GSC2	CAP20	138	Unknown
MEP2	—	AGP1	127	Both activated by low nitrogen levels
—	HAP1	CYT1	115	CYT1 is directly regulated by HAP1
GLK1	—	MSF1	95	Co-induced in mitochondrial mutant
SPO13	—	INO2	94	Both involved in cell division (mitosis/meiosis)
HPR5	—	GLG2	86	Co-induced during G2

Table 5.3: Most common gene pairs in results of Mamdani model.

and Wang'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 for known gene relationships. The gene relationships of the HAP1 regulatory network, examined in [8], were found to have similar error and variance scores as they had in [8]. 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 gene relationships. The most common pairs of genes were found and are summarized in Table 5.3. 'A' denotes the activator gene, 'B' denotes the repressor gene, and 'C' denotes the target. Most of the relationships between the genes were obtained by the Proteome YPD database [21, 22]

One thing that is apparent by the most common pairs is that the Mamdani model extracts many co-regulated pairs of genes. There is no known causal relation between the two, but they appear to rise and fall with similar profiles of expression. With the use of the *min* operator for fuzzy conjunction, it is more likely that changes in a single model input will not change the output. Thus, it is more likely that a particular gene's expression timeseries will have little effect on the output

compared to another. Thus, we are faced with increasing likelihood that frequently expressed pairs are in fact co-regulated and do not have a causal relationship.

5.3 Developing More Complex Models

Because many coactivation or corepression complexes consist of many proteins, time and computational resources prohibited the search of all possible combinations of large numbers of genes. Instead, we checked the results of a model with 2 activators and 1 repressor. If any coactivator complexes are active, pairs of member genes should appear in the coactivator positions.

5.3.1 Model Validation

Unfortunately, only six known coactivators (TSM1, SNF5, SWI1, SWI3, SRC1, and PGD1) are available in the data to be analyzed, most of which are involved in different coactivator complexes. Since the algorithm can only analyze genes whose expression levels have significantly changed (thus having "High" and "Low" expression levels), many coactivators were left out due to relatively constant expression over time. They may all be at a "High" or "Low" expression level, thus completing the complex, but we cannot discern that from the information given.

Our only methods of validation were to check transcription factor enrichment as well as coactivator enrichment. Again, only 3.97% of the input genes are transcription factors, but 15.47% of the combinations with an error score less than 1500 and variance score less than 20000 contain transcription factors, revealing an increase of a factor of 3.9. Only 0.317% of the input genes are known to be coactivators, but they appear in 0.822% of the results with the same cutoffs. Of the six coactivators in the input data, only two are a part of the same coactivator complex; SWI1 and SWI3 are part of the SWI/SNF complex [24]. Both often appear in the same gene combinations, but with one as a coactivator and the other as a target.

Since the known targets of the SWI/SNF complex are not in the input set, we cannot find many combinations where both serve as coactivators. As mentioned in the previous section, it is possible for coregulated genes to appear in activator and target positions provided that other activators and repressors have minimal effect on the model output.

More data will be needed to prove the effectiveness of general model validation. A decrease in signal noise will also be beneficial; it will allow us to analyze genes with lower factors of change over the timeseries since we can be assured that the change is real and not a factor of noise.

5.3.2 Effects of Clustering to Improve Run Time

As expected, the time required for the analysis of a model with three input genes is significantly higher than that of the two input model; a three-input model takes approximately 75 hours with code optimizations on our test systems where a two-input model took less than an hour. Therefore, saving computation time through clustering is even more important. The time requirements also made it difficult to perform the same number of experiments on the general model. As such, our results are not as complete for the general model. We performed tests similar to those performed on the two input model (cluster combination percentile cutoffs and maximum score cutoffs). Our results for the cdc28 dataset can be found in Figures 5.16 and 5.17. Figure 5.17 can be considered a cross-section of graphs in Figures 5.5 - 5.7 at the point where the number of clusters is 15.

With the percentile cutoff method, we see similar behavior to the simple model in the response to clustering. However, time savings are significantly reduced; a 50% percentile cutoff only saves 30% run time. Using this method with the general model reveals the same problem as the simple model: there is no *a priori* knowledge of the optimal cutoff point.

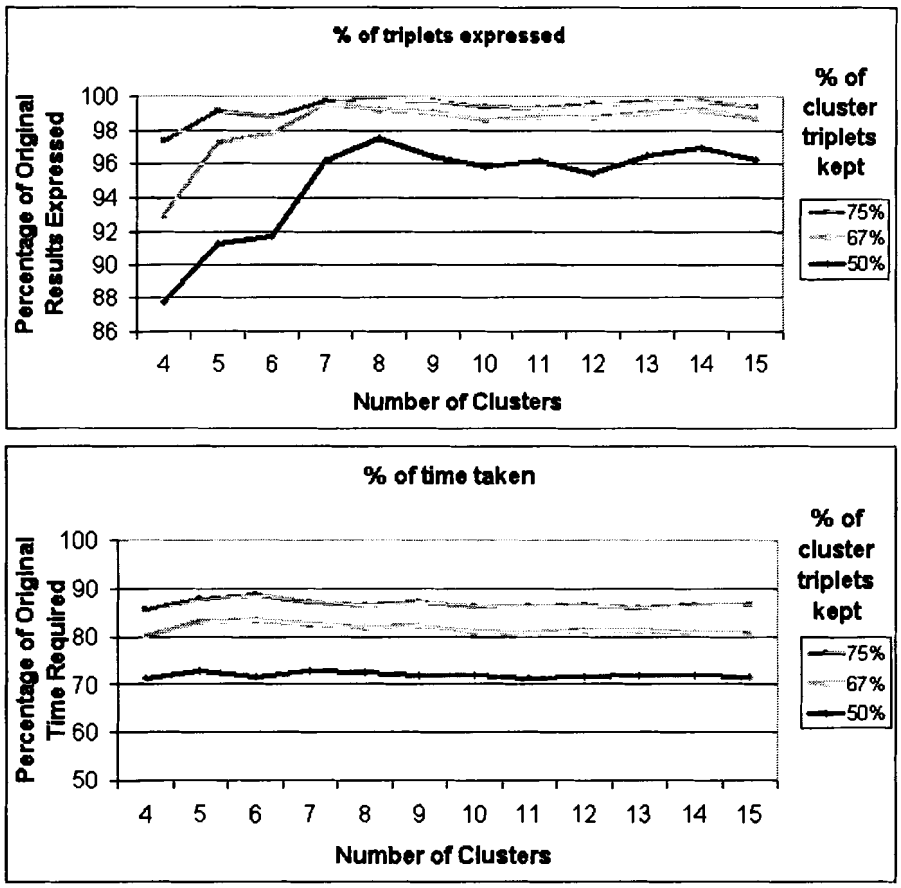


Figure 5.16: Results obtained and time required for clustering method of fuzzy analysis for datasets in [9] using cluster error percentile cutoffs in a 2 activator model

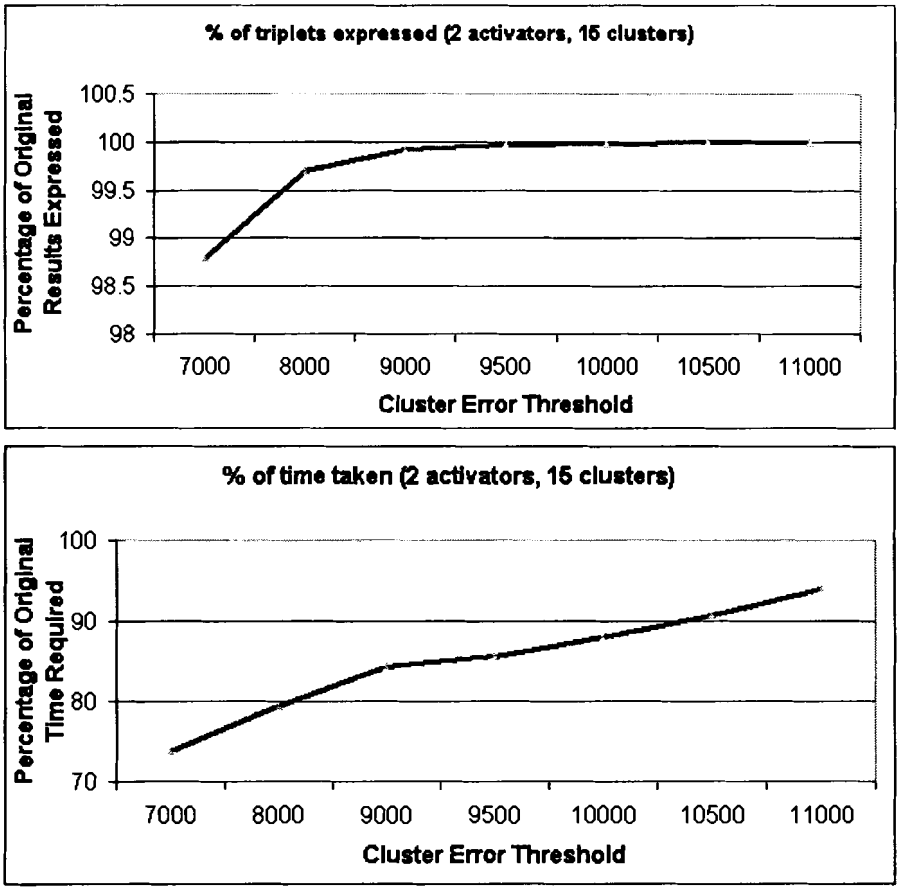


Figure 5.17: Results obtained and time required for clustering method of fuzzy analysis for datasets in [9] using error cutoff thresholds in a 2 activator model

The cluster error threshold method fares better. The 99.9% point is slightly higher than with the simple model (around 9000), but there appears to be a definite point. The time savings is reduced to 20%, but we are still able to find the optimum point. It should also be considered that 20% of 75 hours (the time required to obtain the results of the cdc28 dataset for an error cutoff of 1500 and a variance cutoff of 20000, which does not encapsulate the entire set of results we would want to examine) is quite significant and will be even more beneficial with increased cutoff limits or numbers of inputs.

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

We now draw the following conclusions addressing each point made in the introduction:

1. The use of clustering as a preprocessing method can save a significant amount of processing time. By using a large number of clusters and setting a model-dependent error threshold for cluster combinations, we can effectively obtain the same results as the original algorithm with significantly reduced run time. Empirical evidence shows that the optimal cluster error threshold is linear and predictable. The amount of time saved while still obtaining 99.9% or more of the results is dependent upon the dataset, but ranges from 20-50%

2. Altering the methods of applying the fuzzy model can produce valid results that are less vulnerable to noise. In particular, the Mamdani model is quite resilient to noise and (on average) only adds a constant offset to error scores. This allows the user to merely set a higher error score cutoff to obtain all valid results. However, more testing and validation needs to be before concluding that the Mamdani model is better at predicting known relationships.

3. The general model proposed in this thesis shows promise as a valid model. The results generated by the model enrich the presence of both transcription factors (which are enriched by a factor of 3.9) and known coactivators (which are enriched by a factor of 2.78). Unfortunately, validation could not be completed due to a lack of known coactivators in the dataset. More data will be needed to fully validate the model. The general model also benefits

from using clustering as a preprocessing method; the run time for the cdc28 dataset is reduced by about 17%. As the number of inputs increases, even more time may be saved, even if the relative savings decreases somewhat, as is the case with our results.

We can propose an overall improved method for the algorithm as follows:

1. Perform gene selection and normalization as performed by Woolf and Wang [8].
2. Run a low-pass filter through the data to downplay minor variations and allow for better clustering.
3. Create self-organizing maps for the filtered dataset. Increase the number of clusters until the standard deviation between a node and nearby clusters does not decrease much with an increase in the number of clusters.
4. Reintroduce the unfiltered data into the resulting dataset.
5. Run the fuzzy algorithm using the cluster node expression profiles as gene expression data.
6. Run the modified algorithm (using the Mamdani model), which eliminates combinations of genes whose corresponding clusters do not fit the model well.

6.2 Future Work

While Woolf and Wang's algorithm has been expanded to run faster and increase its robustness, there is still more work to do to make the fuzzy algorithm a proper tool for gene expression analysis. There are still several tasks that need to be completed.

First, there must be further validation of the model as identifying valid gene relationships. While the Mamdani model has been shown to produce valid results, there is no objective way to compare its validity to that of the results of the Woolf and Wang, SAM, or Hybrid versions of the model. There needs to be some measurement of the validity of any model that will allow us to select the best possible version of the fuzzy model. A heuristic may be proposed through examining the algorithm's output with a database of all known gene relationships and reporting percentages of the results that are known causal relationships, known coregulated groups of genes, etc. A fitness score can be established using the number of valid causal relationships found versus the number of invalid causal relationships and the error scores assigned to each. Care must be taken with analyzing percentages, however; there may be a large number of valid relationships found between genes with unknown function that would not be in any database. Thus, absolute numbers or ratios of valid to invalid relationships in known genes might be a better assessment. Other methods of comparing genetic network models have been proposed by Wessels *et al* [26] that may serve as a good starting point for model analysis.

Second, the validity of the general model must be tested further. Due to time and computational constraints, our analysis was limited to one form of the general model (2 activators, 1 repressor) and one dataset. In that dataset, there were few known coactivators that met the cutoffs of minimum expression level and ratio of change over the timeseries. More tests need to be done with more datasets to see if the general model extracts known relationships.

Third, if the general model is proved to be invalid, work will need to be done to find what the general model should be. Development of a better general model may be refined through the use of neuro-fuzzy networks with gene expression data from known sets of coactivators and/or corepressors.

Finally, a data mining method should be developed to extract results of interest from the listing of low-error gene combinations. One cannot always draw the conclusion that a combination of genes has a causal relationship from one microarray experiment. Analysis of the results of the fuzzy algorithm on several different experiment timeseries would further reveal the likelihood of a certain relationship; if the same combination of genes fits the model in numerous timeseries, it is more likely that the genes are related. Work must also be done to identify the difference between causal relationships and groups of genes that are simply coregulated. As shown in the validation of the Mamdani models, there are many low-error combinations that do not imply causal relationships but imply relationship through being coregulated. Clustering may also be able to help in data mining. If an activator (or group of activators) belong to the same cluster as the target (or are in an adjacent cluster), the repressor (or repressors) may have little effect on the fuzzy model and may thus be not as likely to reflect a causal relationship; the genes may simply be coregulated. Other data mining methods may be proposed and validated to help make more sense of the results.

REFERENCES

[1] M. Schena, D. Shalon, R.W. Davis, P.O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467-470, 1995

[2] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, 95:14863-14868, 1998

[3] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, T. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the Natlional Academy of Sciences USA*, 96:2907-2912, 1999

[4] P. D'haeseleer, X. Wen, S. Fuhrman, R. Somogyi. Linear modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing*, 4:112-123, 1999

[5] D.C. Weaver, C.T. Workman, G.D. Stormo Modeling regulatory networks with weight matrices. *Pacific Symposium on Biocomputing*, 3:112-123, 1999

[6] T. Chen, H. He, and G. Church. Modeling gene expression with differential equations. *Pacific Symposium on Biocomputing*, 4:29-40, 1999

[7] S. Liang, S. Fuhrman, R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing* 3:18-29, 1998

[8] P.J Woolf, Y. Wang. A fuzzy logic approach to analyzing gene expression data. *Physiological Genomics* 3:9-15, 2000

[9] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart *et al.* Genome-Wide Analysis of Cell Cycle-Dependent Transcription. *Molecular Cell*, 2:65-73, 1998

[10] Anonymous. GEM Microarray Reproducibility Study. Incyte Pharmaceuticals, Inc, 1999

[11] M.K. Kerr, M. Martin, G.A. Churchill. Experimental design for gene expression microarrays *Biostatistics*, 2:183-201, 2001

[12] B. Lewin. *Genes*, Oxford University Press, 1997

[13] National Human Genome Research Institute Office of Science Education and Outreach. `http://www.nhgri.nih.gov/DIR/VIP/Glossary/Illustration/nucleotide.html` and `http://www.nhgri.nih.gov/DIR/VIP/Glossary/Illustration/dna.html`; accessed October 28, 2001.

[14] National Human Genome Research Institute Office of Science Education and Outreach. `http://www.nhgri.nih.gov/DIR/VIP/Glossary/Illustration/mrna.html`; accessed October 28, 2001

[15] Kerr M.K., Martin M., Churchill G.A. Analysis of variance for gene expression microarray data. *Journal of Computational Biology* 7:819-837, 2000

[16] 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

[17] Z. Szallasi. Genetic network analysis in light of massively parallel biological data acquisition. *Pacific Symposium on Biocomputing.* 4:5-16, 1999

[18] T. Kohonen. The self-organizing map. *Proceedings of the Institute of Electrical and Electronics Engineers*, vol. 78:1464-1480, 1990

[19] E.H. Mamdani, S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Machine Studies*, 7(1), 1975

[20] B. Kosko. *Fuzzy Engineering*, Prentice Hall, 1997

[21] M.C. Costanzo, J.D. Hogan, M.E. Cusick, B.P. Davis, A.M. Fancher, P.E. Hodges, P. Kondu, C. Lengieza, J.E. Lew-Smith, C. Lingner, K.J. Roberg-Perez, M. Tillberg, J.E. Brooks, J.I. Garrels. The Yeast Proteome Database (YPD) and Caenorhabditis elegans Proteome Database (WormPD): comprehensive resources for the organization and comparison of model organism protein information. *Nucleic Acids Research* 28(1):73-76, 2001. Abstract

[22] M.C. Costanzo, J.D. Hogan, M.E. Cusick, B.P. Davis, A.M. Fancher, P.E. Hodges, P. Kondu, C. Lengieza, J.E. Lew-Smith, C. Lingner, K.J. Roberg-Perez, M. Tillberg, J.E. Brooks, J.I. Garrels. YPD™, PombePD™, and WormPD™: model organism volumes of the BioKnowledge™library, an integrated resource for protein information. *Nucleic Acids Research* 29(1):75-79, 2001. Abstract

[23] M.H. Peters, J.P. Beltzer, G.B. Kohlhaw. Expression of the yeast LEU4 gene is subject to four different modes of control. *Archive of Biochemistry and Biophysics* 276(1):294-298, 1990

[24] J. Cote, J. Quinn, J.L. Workman, C.L. Peterson. Stimulation of GAL4 derivative binding to nucleosomal DNA by the yeast SWI/SNF complex. *Science* 265(5168):53-60, 1994

[25] R. Reynolds, H. Ressom, M. Musavi. Use of clustering to improve performance in fuzzy gene expression analysis. *International Joint Conference on Neural Networks.* 2738-2743, 2001

[26] L.F.A. Wessels, E.P. Van Someren, M.J.T. Reinders. A Comparison of Genetic Network Models. *Pacific Symposium on Biocomputing.* 6:508-519, 2001

APPENDIX A

Monte Carlo Error Simulations For Fuzzy Models

All Monte Carlo error simulations of the four fuzzy models (Woolf and Wang, Mamdani, SAM, and Hybrid) with noise margins from 5-30% can be found below in Figures A.1 - A.12.

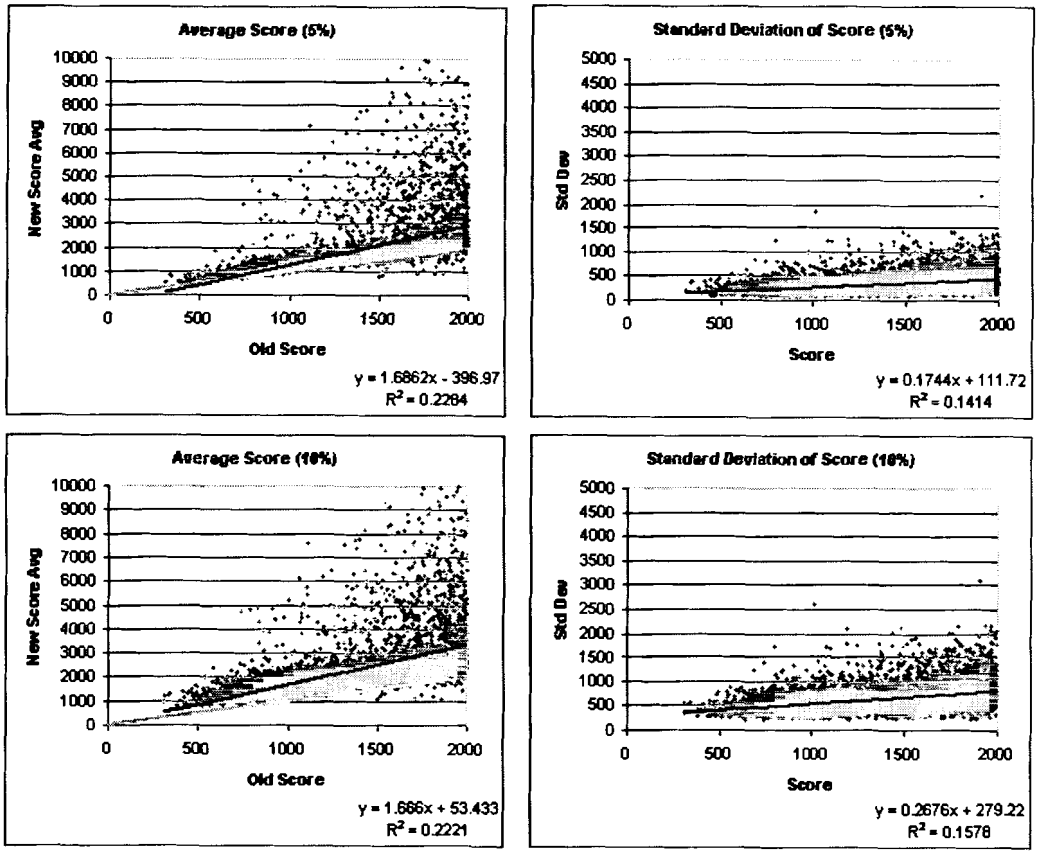


Figure A.1: Monte Carlo error simulations for the Woolf & Wang model (error 5-10%)

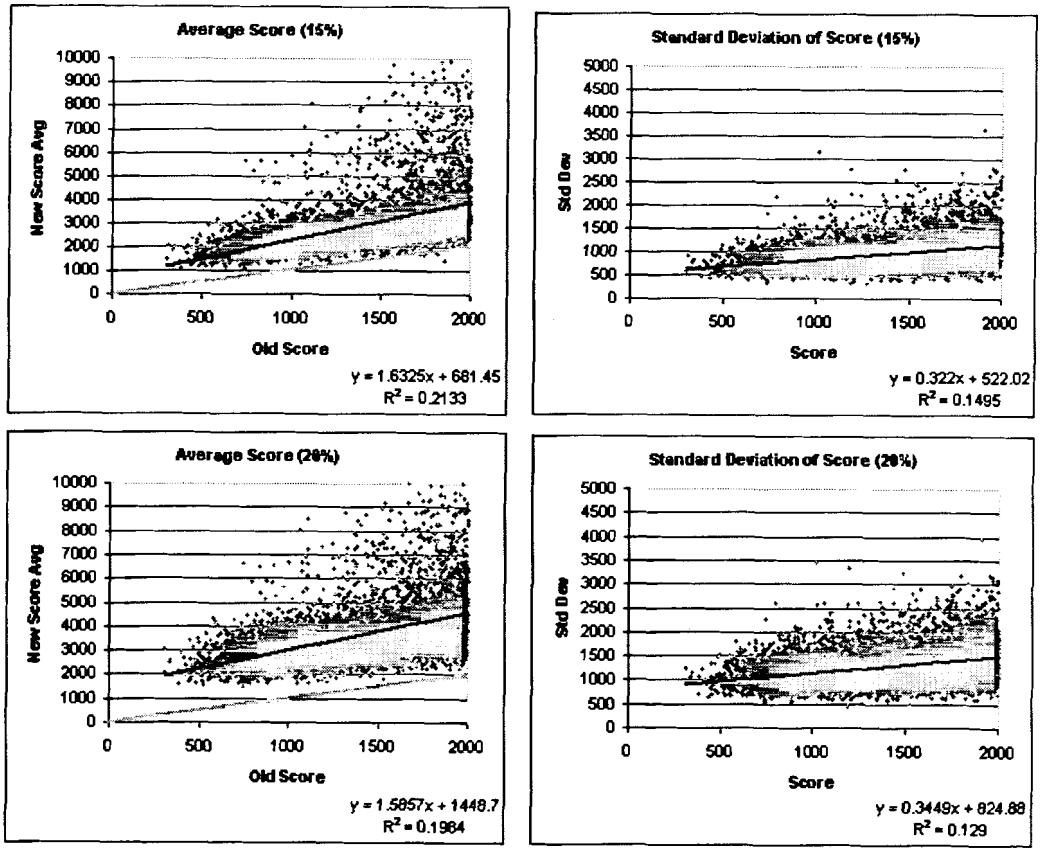


Figure A.2: Monte Carlo error simulations for the Woolf & Wang model (error 15-20%)

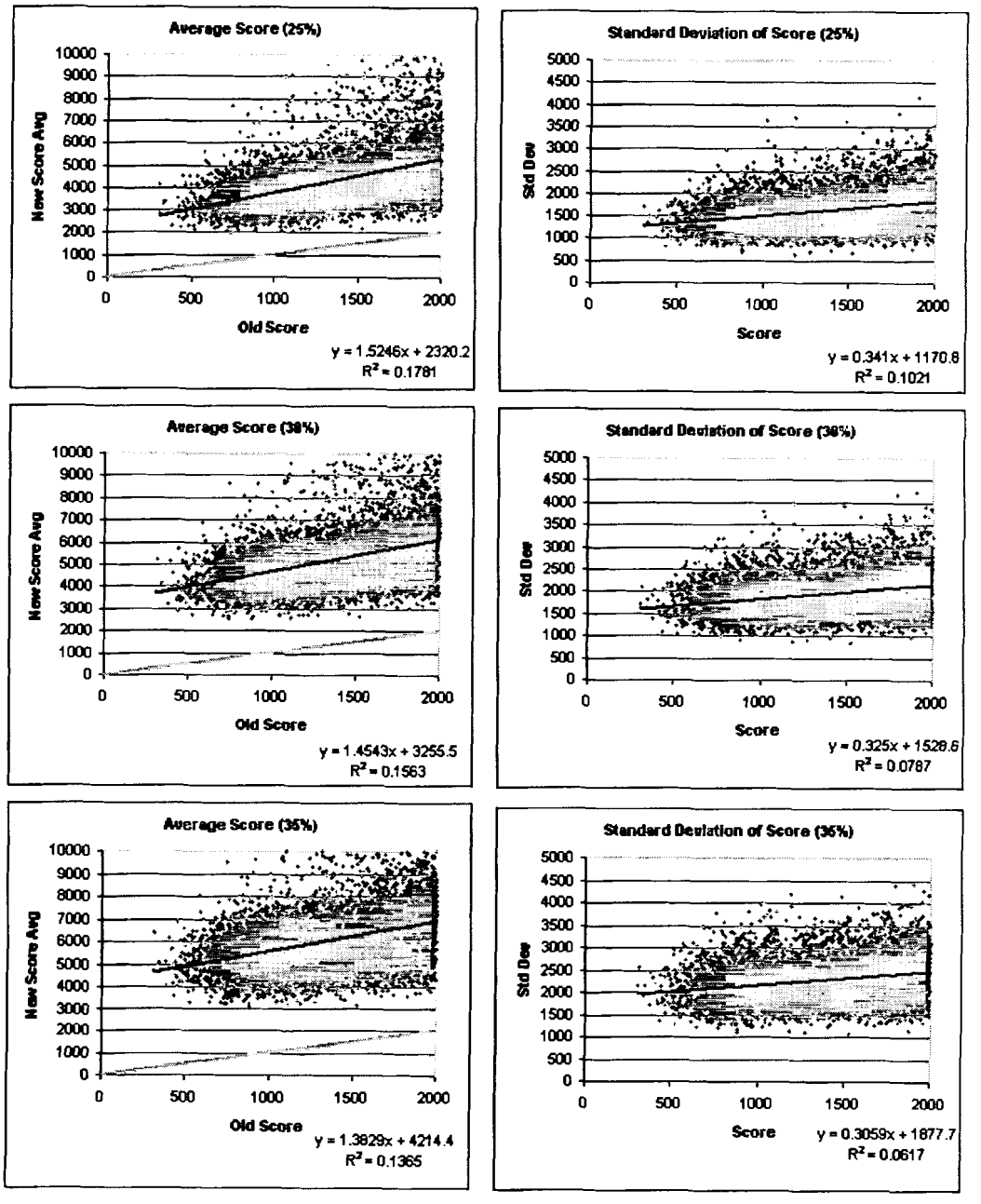


Figure A.3: Monte Carlo error simulations for the Woolf & Wang model (error 25-35%)

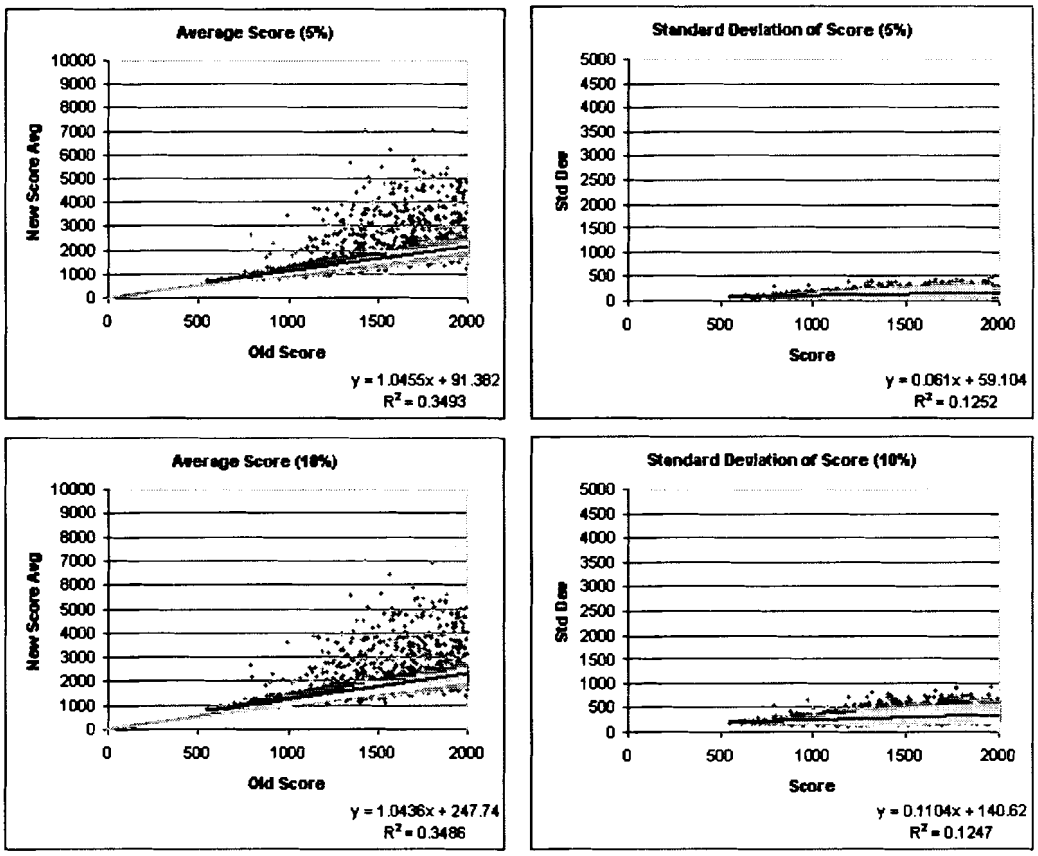


Figure A.4: Monte Carlo error simulations for the Mamdani model (error 5-10%)

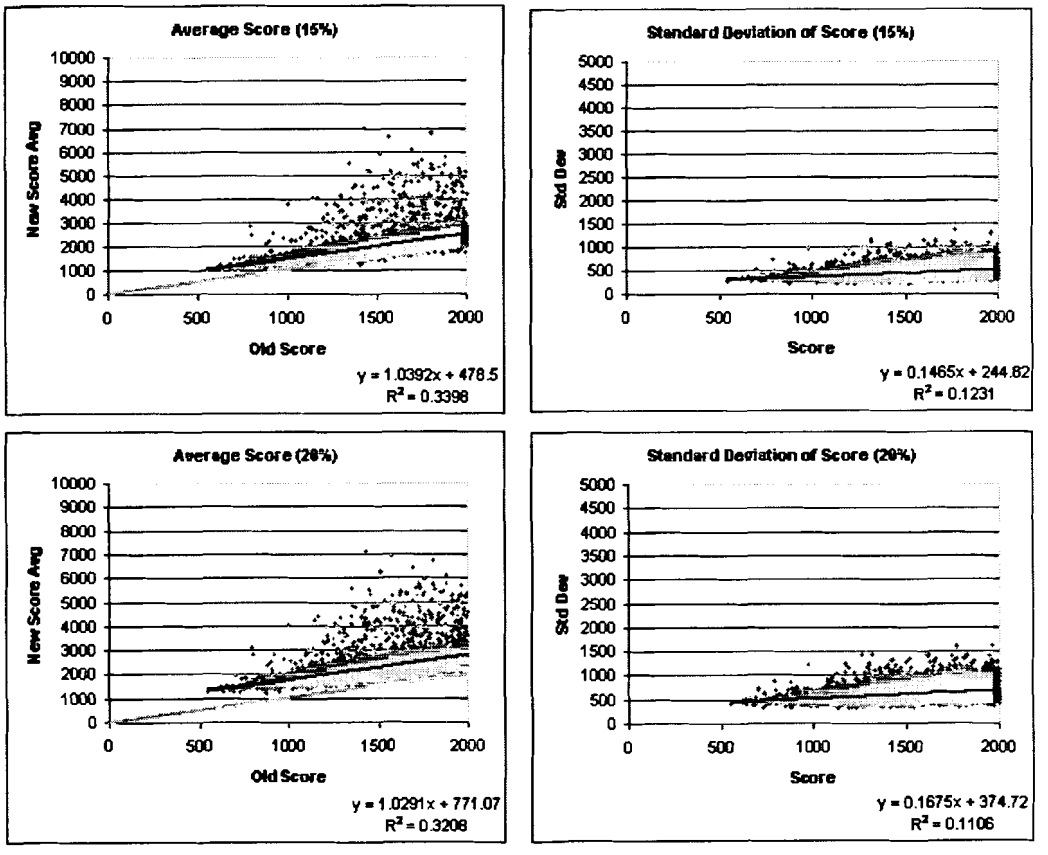


Figure A.5: Monte Carlo error simulations for the Mamdani model (error 15-20%)

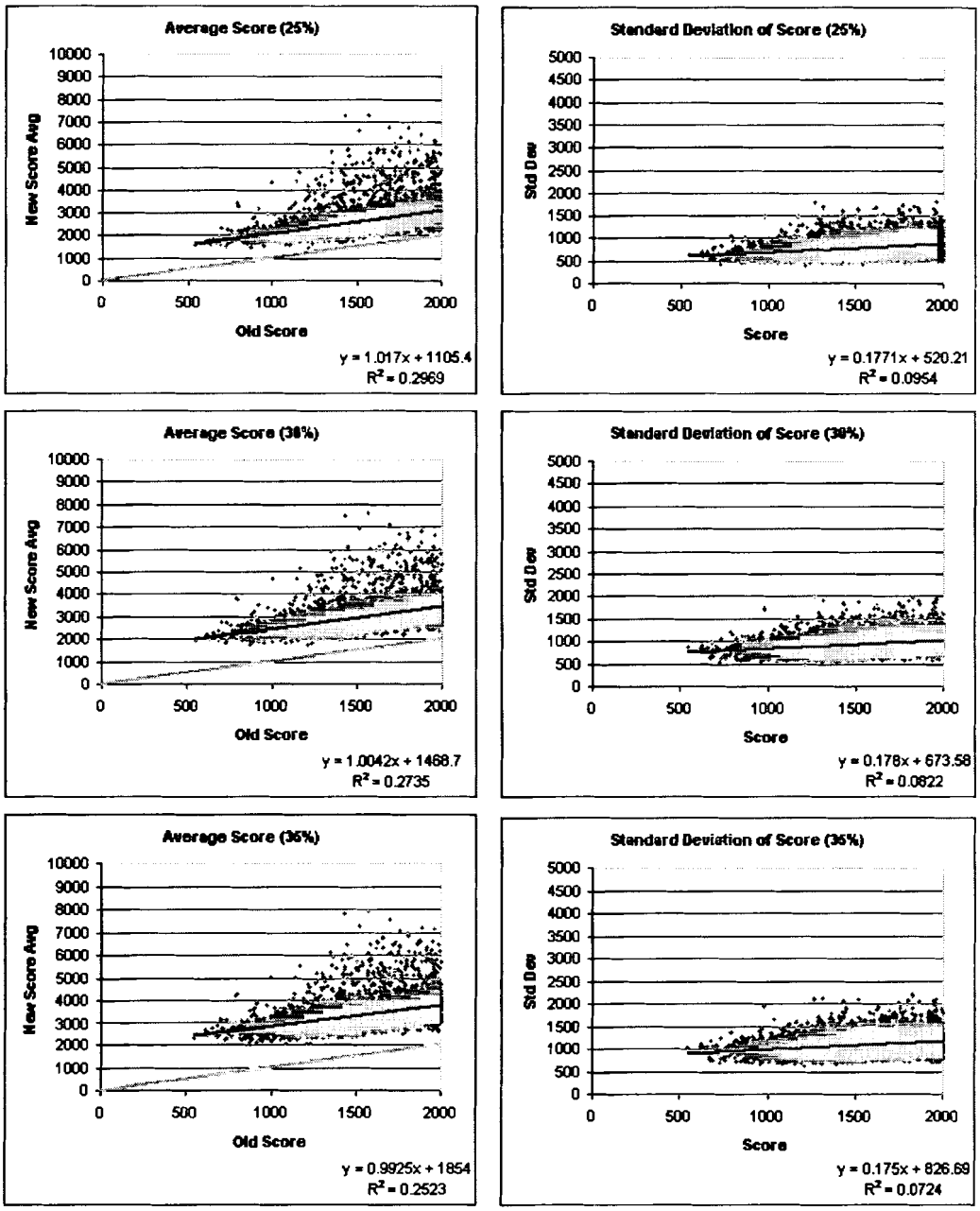


Figure A.6: Monte Carlo error simulations for the Mamdani model (error 25-35%)

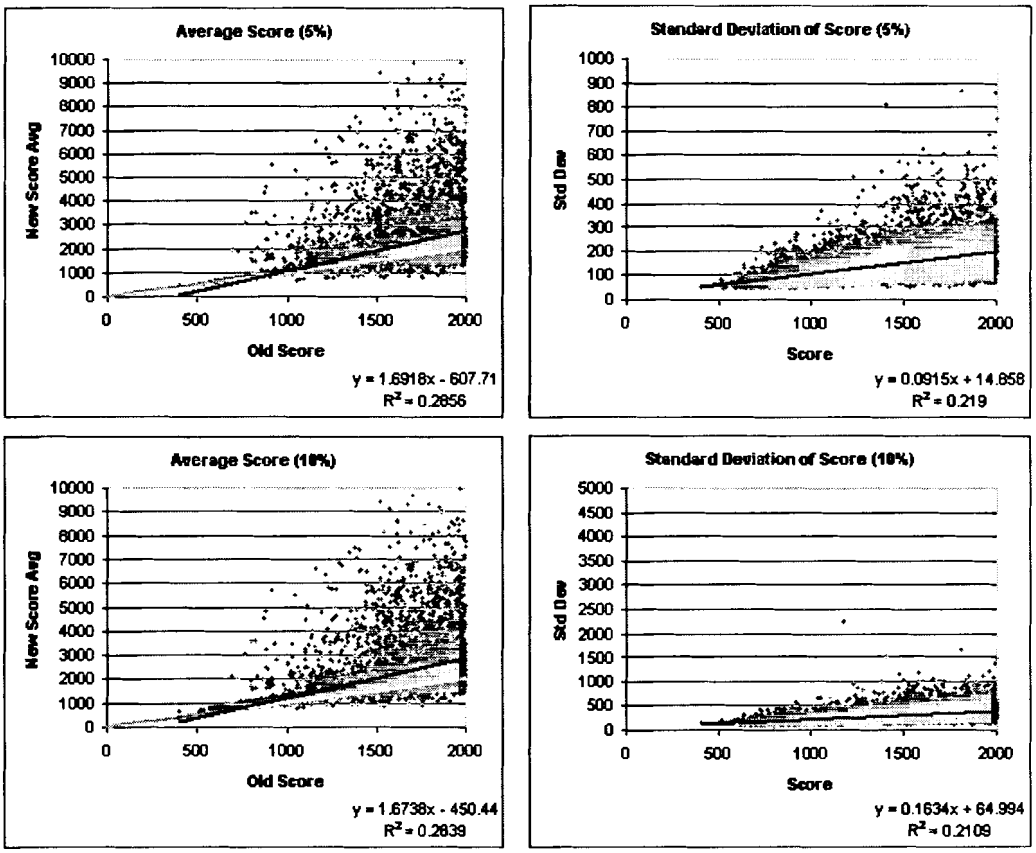


Figure A.7: Monte Carlo error simulations for the Standard Additive Model (error 5-10%)

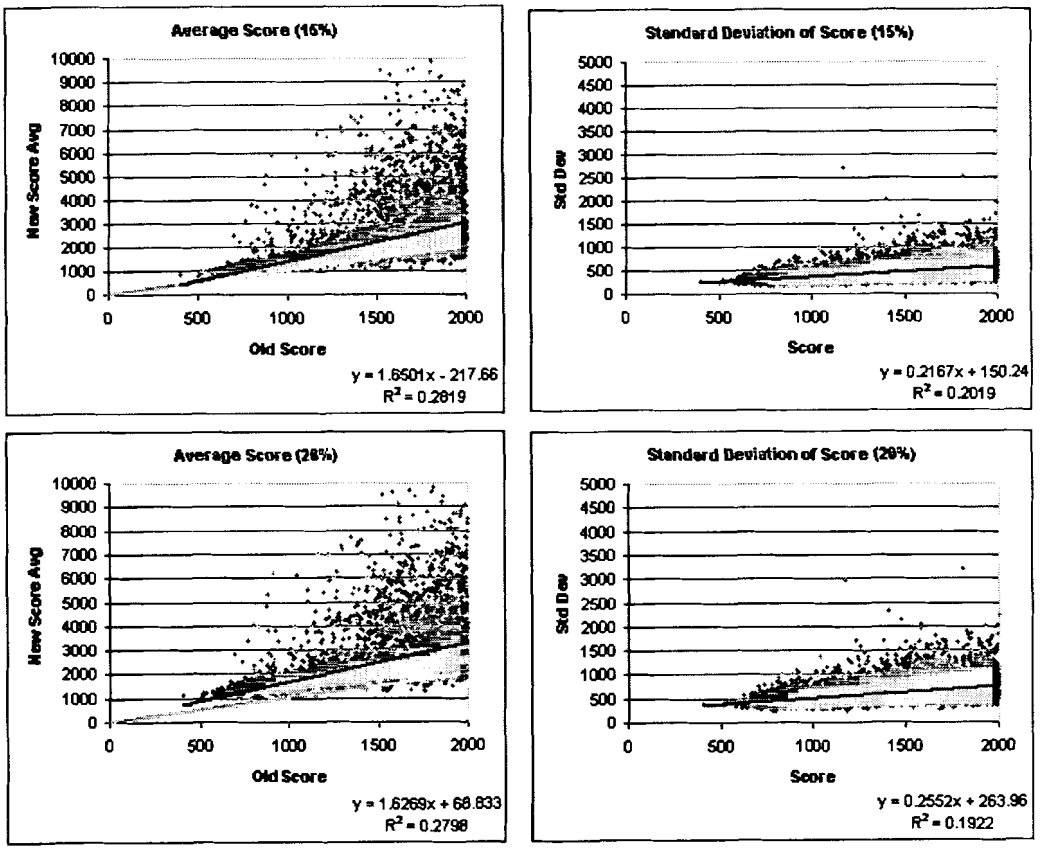


Figure A.8: Monte Carlo error simulations for the Standard Additive Model (error 15-20%)

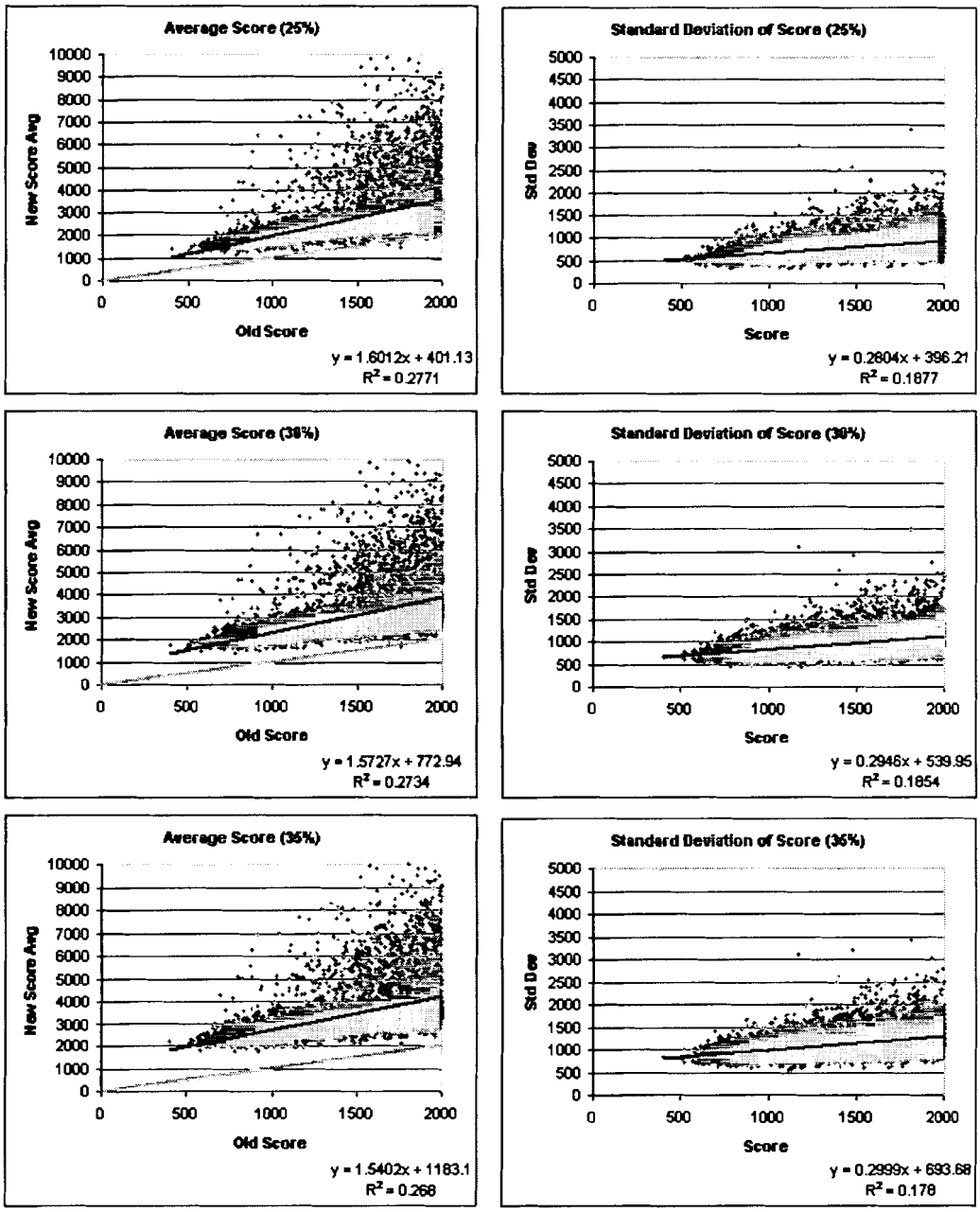


Figure A.9: Monte Carlo error simulations for the Standard Additive Model (error 25-35%)

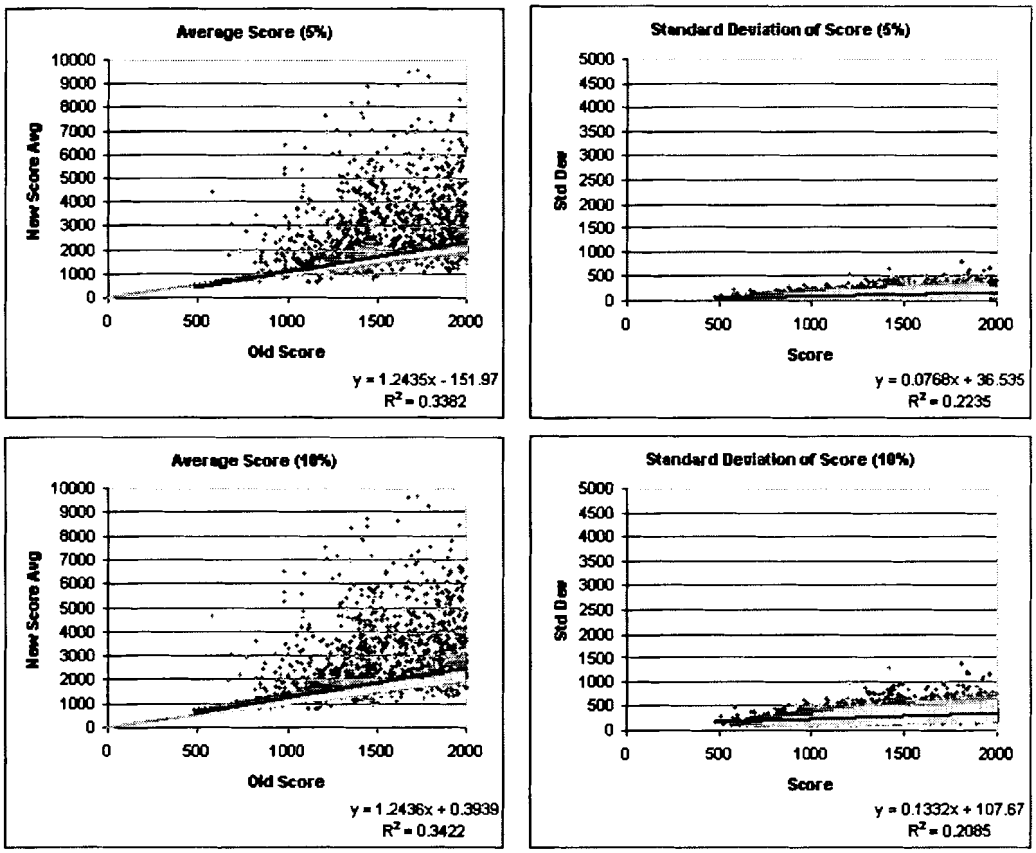


Figure A.10: Monte Carlo error simulations for the Hybrid model (error 5-10%)

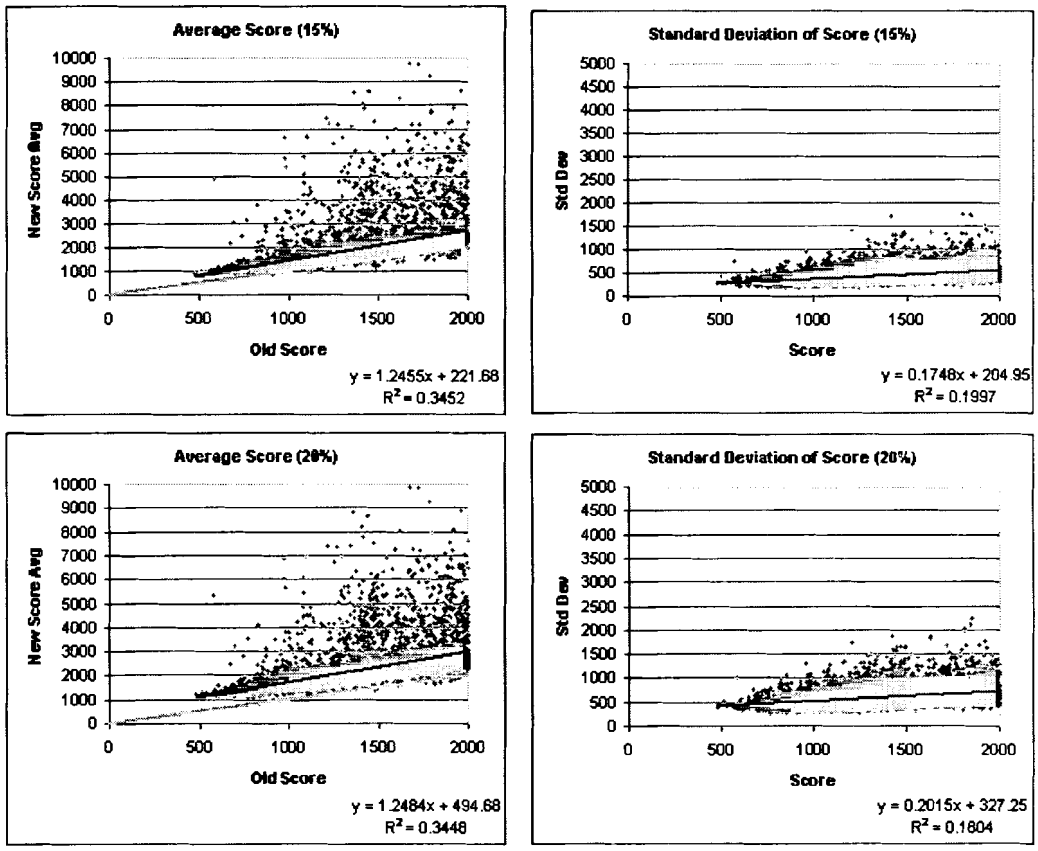


Figure A.11: Monte Carlo error simulations for the Hybrid model (error 15-20%)

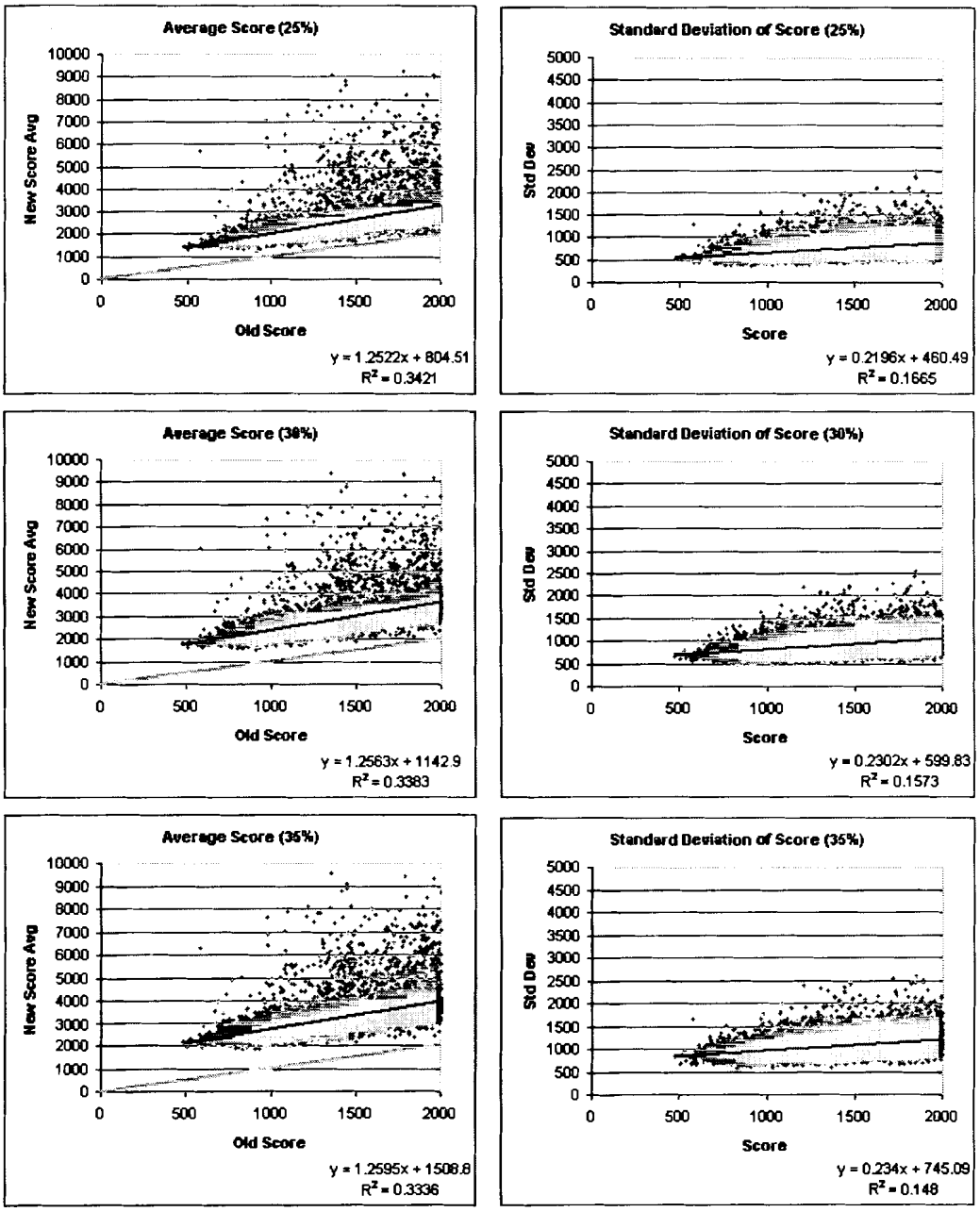


Figure A.12: Monte Carlo error simulations for the Hybrid model (error 25-35%)

Biography of the Author

Robert Reynolds was born in Bangor, Maine on January 31, 1978. He graduated from Brewer High School in 1996.

He entered the University of Maine in 1996 and obtained his Bachelor of Science in Computer Engineering in 2000. His undergraduate work included work in the Instrumentation Research Lab, where he developed Internet software and embedded systems.

In May 2000, he was enrolled for graduate study in Computer Engineering at the University of Maine and served as Research Assistant in the Intelligent Systems Laboratory. His current research interests include fuzzy logic and neural networks.

Robert is an Eagle Scout. He is a member of IEEE, Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi, Kappa Kappa Psi, and the University of Maine Marching, Pep, and Concert Bands. Robert is a candidate for the Master of Science degree in Computer Engineering from the University of Maine in December, 2001.