International Journal of Advanced Research in

Computer Science Engineering and Information Technology

Volume: 3 Issue: 1 26-Jun-2014,ISSN_NO: 2321-3337

Systematic and Comparative study on Gene Micro Array using Fuzzy Logic Applications

G.Gunasekaran[1], R.Rajeswari [2]

Principal, Meenakshi Engineering College, Chennai,India
Research scholar, St. Peter's Institute of Higher Education & Research, St. Peter's University, Avadi, Chennai 600054.

ABSTRACT *- Gene expression data is one of the most important areas that have emerged in the field of bioscience and medicine. There is a vast amount of data related with the gene expressions. The approach used for mining of the gene expression data is K-means clustering method while performing the retrievals and the computations parallel, thereby decreasing both the processing time and performing mining efficiently. The paper aims at solving the problem from the field of bioscience in the Engineering perspective.*

1. INTRODUCTION

To obtain knowledge from the data, explore relationships between genes, understanding severe diseases and development of drugs for patterns from the databases of large size and high Dimensionality. Information retrieval and data mining are powerful tools to extract information from the databases and/or information repositories. The integrative cluster analysis of both clinical and gene expression data has shown to be an effective alternative to overcome the abovementioned problems. In this paper, we focus on how to improve the searching and the clustering performance in genomic data from commonly used clustering techniques. Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

1.1 GENE EXPRESSION DATA

Gene expression data clustering starts from the Microarray experiments in which the genes of the whole genome are represented on a computer based chip. The information from the microarray is analyzed using the computers.

Microarrays are one of the latest breakthroughs in experimental molecular biology, which allow monitoring of gene expression for tens of thousands of genes in parallel and are already producing huge amounts of valuable data. Analysis and handling of such data is becoming one of the major bottlenecks in the utilization of the technology.

The raw microarray data are images, which have to be transformed into gene expression matrices-tables where rows represent genes, columns represent various samples such as tissues or experimental conditions, and numbers in each cell characterize the expression level of the particular

gene in the particular sample. These matrices have to be analysed further, if any knowledge about the underlying biological processes is to be extracted. In this paper we concentrate on discussing bioinformatics methods used for such analysis.
We briefly discuss supervised and unsupervised data analysis and its applications, such as predicting gene function classes and cancer classification. Then we discuss how the gene expression matrix can be used to predict putative regulatory signals in the genome sequences. In conclusion we discuss some possible future directions.

1.2 DATA MINING

One of the main challenges in classifying gene expression data is that the number of genes is typically much higher than the number of analyzed samples. Also, it is not clear which genes are important and which can be omitted without reduce the classification performance. Many pattern classifications techniques have been employed to analyze microarray data.
For the informatics specialist, a centralized platform can integrate bioinformatics data and applications, providing the flexibility to quickly evolve data processing workflows as new algorithms are made available.
For the bench scientist, data processing workflows are available in an easy-to-use web interface to quickly interpret gene expression results and use powerful text analytics to help identify critical information in the literature.

1.3 FUZZY MINING

A fuzzy logic approach to analyzing gene expression data. Physiol Genomics 3: 9–15, 2000. We have developed a novel algorithm for analyzing gene expression data. This algorithm uses fuzzy logic to transform expression values into qualitative descriptors that can be evaluated by using a set of heuristic rules. In our tests we designed a model to find triplets of activators, repressors, and targets in a yeast gene expression data set. For the conditions tested, the predictions made by the algorithm agree well with experimental data in the literature.
The algorithm can also assist in determining the function of uncharacterized proteins and is able to detect a substantially larger number of transcription factors that could be found at random. This technology extends current techniques such as clustering in that it allows the user to generate a connected network of genes using only expression data.

1.4 CLUSTERING TECHNIQUES

A system of cluster analysis for genome-wide expression data from DNA microarray hybridization is described that uses standard statistical algorithms to arrange genes according to similarity in pattern of gene expression. The output is displayed graphically, conveying the clustering and the underlying expression data simultaneously in a form intuitive for biologists. We have found in the budding yeast Saccharomyces cerevisiae that clustering gene expression data groups together efficiently genes of known similar function, and we find a similar tendency in human data. Thus

patterns seen in genome-wide expression experiments can be interpreted as indications of the status of cellular processes. Also, coexpression of genes of known function with poorly characterized or novel genes may provide a simple means of gaining leads to the functions of many genes for which information is not available currently.

2. RELATED WORK

2.1. GENE EXPRESSION DATA CLUSTERING METHODS

Cluster analysis partitions a given dataset into the set of groups, based on specific characteristics in such a way that the data within a particular group is more similar than the other groups. The disjoint groups formed due to the clustering process are said to be as the clusters. The datasets within the group are more similar to each other while the datasets in the separate groups are less similar. Traditional clustering algorithms consist of (i) partition based method; (ii) level based method; (iii) density based method; (iv) grid based method; and (v) model based method. The gene expression clustering methods are broadly categorized into 2 categories:

2.1.1) Clustering of co-expressed genes based on gene expression pattern.
2.1.2) Subspace clustering

The first category is further sub-divided into two types:

Gene based Clustering:

Gene based clustering regards genes as objects and the samples are treated as features. This clustering technique clusters the total number of genes on the basis of total samples.

Sample based Clustering:

Sample based clustering treats the samples as objects and the genes are regarded as features. This clustering technique clusters the total number of the available samples on the basis of total genes.

Subspace Clustering:

In this clustering technique, the clusters formed by a particular subset of genes are captured across a subset of the samples used. This clustering technique treats genes and samples symmetrically, such that either of the genes or the samples can be treated either as the objects or features. Also the clusters that are generated using subspace clustering algorithms, have usually disjoint feature spaces.

2.2 GENE EXPRESSION IN FUZZY LOGIC

There are three main advantages of applying fuzzy logic to the analysis of gene expression data. First, fuzzy logic inherently accounts for noise in the data because it extracts trends, not precise values. Second, in contrast to other automated decision making algorithms, such as neural networks or polynomial fits, algorithms in fuzzy logic are cast in the same language used in day-to-day conversation. As a result, predictions made using fuzzy logic are easily interpretable and can be extrapolated in predictable ways. Third, fuzzy logic techniques are computationally efficient and can be scaled to include an unlimited number of components. Thus they are able to recognize a large number of biologically important patterns.

In this work we present a fuzzy logic based algorithm for analyzing gene expression data. Using fuzzy logic, we have developed a analysis technique that can identify logical relationships between genes and in some cases even predict the function of an unknown gene. This algorithm was validated using yeast expression data gathered from the Affymetrix GeneChip system. By using yeast gene expression data collected at different time points of the cell cycle, we were able to identify many regulatory elements and their target genes within the cell that work together to maintain and control certain cellular processes. Several cases are validated by available experimental results, including the signaling network controlled by the transcription factors HAP1 and ROX1, which control the transition from anaerobic to aerobic growth. These results suggest that our fuzzy logic technique can indeed find biologically relevant connections between sets of genes, which in turn could help to describe the complex web of interactions that regulate gene expression.

3. FUZZY MINING TECHNIQUES

Fuzzy logic is an algorithm drawn from engineering and other applied sciences to control systems as diverse as washing machines to autofocus cameras (2, 10). It provides a way to transform precise numbers, such as 32.43, into qualitative descriptors, such as “high” in a process called “fuzzification.” Although other techniques can be used to change precise values into discrete descriptors, fuzzy logic provides a systematic and unbiased way to perform this transformation, thereby removing the need for expert knowledge about the system.

For example, is 32.43 a high value? If 32.43 is a measure of the ambient air temperature in degrees celsius, then most people would say that 32.43°C is a high temperature. But this analysis requires our own expert knowledge, which can vary from person to person. Someone from a tropical climate may feel that 32.43°C is a medium temperature, whereas someone from a very cold climate may take 32.43°C as a very high temperature.

When dealing with gene expression data, the problem is even more complicated, because no expert exists to determine what defines a “high” expression level. Using fuzzy logic, the full range of data is first measured and is then broken into discrete subsections based on the observed data. These discrete subsections then provide a qualitative description of the data. Once transformed, this qualitative data can be analyzed using heuristic rules, which in turn generate fuzzy solutions.

For example, the heuristic rule “if high then move fast” takes “high” as a fuzzy input and “fast” as a fuzzy solution.

In another process called "defuzzification," this heuristic solution can be transformed from a qualitative descriptor back into a precise number.

3.1 FUZZY RULES IN DATABASE

A fuzzy rule involves a fuzzy condition and a fuzzy conclusion. The test dataset consists of hundred random genes and it is selected from the whole dataset of 4026 genes with its samples. It is converted to fuzzy values as shown in Table. The genes included in the test dataset are not selected as top genes in informative genes set.
Sample Test Data from Lymphoma Dataset

GENE ID	VALUES	VALUES	VALUES	VALUES
GENE143X	-0.5224	-0.1563	-0.3851	-0.1929
GENE141X	-0.5407	-0.1425	-0.3989	0.4155
GENE3844X	-0.0922	-0.1975	-0.0876	1.0000
GENE1400X	0.0568	0.5622	-0.0327	-0.0373
GENE137X	-0.4858	-0.4675	-0.4080	0.9208

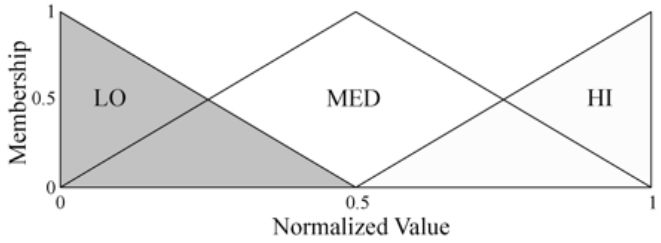
The three lymphoma subtypes are identified by a specific fuzzy rule by assigning intermediate value ranges. A single gene, 2GC and 3GC classifies all genes included in the test dataset and individual count of the relevant lymphoma subtypes is displayed as it is shown in Figure. The subtypes not classified under mentioned lymphoma subtypes is grouped under other subtypes.

Classifying test dataset using fuzzy rule

Classifying Gene	**Gene to be Classified**	**Lymphoma Subtypes**			
		DBLCL	**FL**	**CL**	**Other**
Single Gene					
2GC	Gene x1…	Count(DBLCL) Count(Other)	Count(FL)	Count(CLL)	
3GC	Gene xn	…n	…n	…n	…n

3.2 FUZZY LOGIC ALGORITHM

In analyzing genetic expression data, the data is transformed from crisp values to fuzzy values in a process called "fuzzification." Data is fuzzified by first normalizing the data from 0 to 1, and then the normalized value is broken up into various membership classes. For example, Fig. Shows the three fuzzy sets used in this algorithm, "HI," "MED," and "LO" as a function of the normalized value. For a normalized value of 0.25, the fuzzy value is 0.5 LO, 0.5 MED, and 0 HI; or said another way, 0.25 is 50% low, 50% medium, and 0% high.
The three fuzzy sets HI, MED, and LO were chosen after manually examining expression data and finding that the abundance of most transcripts was high, medium or low. Other schemes that include a different number or shape of fuzzy sets could also be used to better represent the data; however, these modifications tend to make the analysis less general and more complex and therefore were not pursued in this study.



After the data is fuzzified, triplets of data are compared using a set of heuristic rules in the form of a decision matrix. Triplets were defined as the expression values of three different proteins (A, B, and C) all taken at the same time point in the yeast growth cycle time series. Fuzzified values of A and B are entered into this matrix, and at points where their predictions overlap, a score is generated as the fuzzified value of predicted C. This form of a fuzzy value for C that can be defuzzified back into a crisp number.

The predicted expression values of C for each time point in the time series were calculated. Then, the entire series of the predicted values was compared with that of the observed C measurements. For each triplet, the agreement with the assertion in the rule table can be calculated based on square of the residual, r^2, between the calculated C and the observed C. Those triplets that have a low r^2 value fit the assertion better and as such are reported with higher confidence. In our initial screen, we only accepted those triplets with $r^2 < 0.015$, corresponding to an average error of 3% or less, well below the error associated with the expression data that was estimated as 15%

IF "A" IS \ IF "B" IS	HI	MED	LO
HI	"C" is MED	"C" is HI	"C" is HI
MED	"C" is LO	"C" is MED	"C" is HI
LO	"C" is LO	"C" is LO	"C" is MED

In some cases the data set of A and B fail to properly explore the decision matrix (i.e., A is almost always high, and B is almost always low), thus a second score called the variance was also assigned to the data set. The variance is defined as the statistical variance between the total hits in each box on the decision matrix.

If the data set hits are evenly distributed throughout the decision matrix, then the variance score is low, and the resulting predictions are credible. However, if the data set is poorly distributed, then the variance will be high and the predictions may or may not be believable because of the lack of combinations of A and B tested. For the initial screen, only those A/B pairs with a variance of 1.5 or less were chosen.
To get an overall idea of how well the assertion fits the data, the r^2 value and the variance are multiplied and scaled by a factor of 100,000 to give an overall score. Thus triplets with low r^2 values and low variance will have the lowest score and also should be the most credible statements. Other data that are only low in one parameter may be filtered out because either the fit is too poor or the data set is biased.

4. GENE DATA PATTERNS

One of the main challenges in classifying gene expression data is that the number of genes is typically much higher than the number of analyzed samples. Also, it is not clear which genes are important and which can be omitted without reducing the classification performance.
Many pattern classification techniques have been employed to analyze microarray data. For example, Golub *et al.* [1] used a weighted voting scheme, Fort and Lambert-Lacroix [3] employed partial least squares and logistic regression techniques, whereas Furey *et al.* [4] applied support vector machines. Dudoit *et al.* [5] investigated nearest neighbor classifiers, discriminant analysis, classification trees, and boosting, while Statnikov *et al.* [6] explored several support vector machine techniques, nearest neighbor classifiers, neural networks, and probabilistic neural networks. In several of these studies, it has been found that no one classification algorithm is performing best on all datasets (although for

several datasets, SVMs seem to perform best), and hence, the exploration of several classifiers is useful.
Similarly, no universally ideal gene selection method has yet been found as several studies [6], [7] have shown. It should also be pointed out that classification tech- niques like support vector machines are, in general, treated as a black box, which, apart from the actual classification, provide little extra information.
In this paper, we present a fuzzy-rule-based classification sys- tem applied to the analysis of microarray expression data and show, based on a series of experiments, that it affords good classification performance for this type of problem.

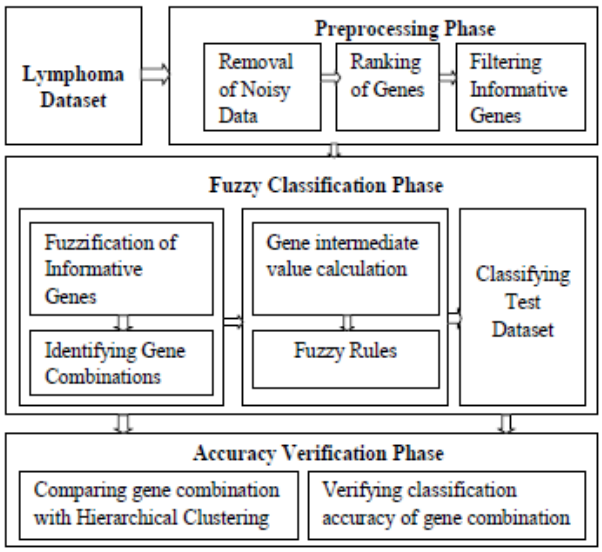
4.1 FUZZY-RULE-BASED CLASSIFICATION

While in the past, fuzzy-rule-based systems have been ap- plied mainly to control problems [10], more recently, they have also been applied to pattern classification problems [11], [12].
Various methods have been proposed for the automatic gen- eration of fuzzy if-then rules from numerical data for pattern classification and have been shown to work well on a variety of problem domains [13]–[15]. Pattern classification typically is a supervised process where, based on a set of training samples with known classifications, a classifier is derived that performs automatic assignment to

Classes based on unseen data. Let us assume that our pattern clas- sification problem is an n-dimensional problem with C classes (in microarray analysis, C is often 2) and m given training patterns xp = (xp1, xp2, . . . , xpn), p = 1, 2, . . . ,m. Without loss of generality, we assume each attribute of the given training patterns to be normalized into the unit interval [0, 1]; i.e., the pattern space is an n-dimensional unit hypercube [0, 1]n. In this study, we use fuzzy if-then rules of the following type as a base of our fuzzy rule-based classification systems:
Rule Rj : If x1 is Aj1 and . . . and xn is Ajn then Class Cj with CFj, j = 1, 2, . . . ,N (1) whereRj is the label of the j-th fuzzy if-then rule,Aj1, . . . ,Ajn are antecedent fuzzy sets on the unit interval [0, 1], Cj is the consequent class (i.e., one of the C given classes), and CFj is the grade of certainty of the fuzzy if-then rule Rj . As antecedent fuzzy sets, we use triangular fuzzy sets as in Fig. 1, where we show the partitioning of a variable into a number of fuzzy sets.
Our fuzzy-rule-based classification system consists of N lin- guistic rules each of which has a form as in (1). There are two steps in the generation of fuzzy if-then rules: specification of antecedent part and determination of consequent class Cj and the grade of certainty CFj . The antecedent part of fuzzy if-then rules is specified manually. Then, the consequent part (i.e., con- sequent class and the grade of certainty) is determined from the given training patterns [16]. It is shown in [17] that the use of the grade of certainty in fuzzy if-then rules allows us to gener- ate comprehensible fuzzy-rule-based classification systems with high classification performance.

4.2 FORMULATION AND METHODOLOGY

The proposed framework Microarray Gene Classification using Fuzzy Logic (MGC-FL) given in Figure 1 is used to find informative gene combinations and to classify gene combinations belonging to its relevant subtype by using fuzzy logic. In the initial phase the noisy data is removed and genes are ranked based on their statistical scores.



The highly informative genes are filtered based on ranking of genes. In the classification phase informative genes are fuzzified and identified for 2-gene and 3-gene combinations. The intermediate value for gene combination is calculated to classify gene lymphoma subtypes by using fuzzy rules. In the final phase top gene combinations are compared with clustering and the classification accuracy of gene combinations is analyzed.

4.2.1 DATASET

Microarrays is one of the latest breakthroughs in experimental molecular biology, which allow monitoring of gene expression for tens of thousands of genes in parallel and are producing huge amounts of valuable data.

GENE ID	NAME	VALUES	VALUES	VALUES
GENE 3129X	Autocrine motility factor Receptor Clone=1072873	-0.3000	0.3000	0.5900
GENE 3126X	2B Catalytic subunit	-0.2200	-1.2100	1.4100
GENE 3072X	APC Clone=125294	-0.0400	0.1500	0.6800
GENE 3067X	Probable ATP Clone=1350869	0.4100	-0.3400	-0.1800
GENE 4006X	SRC-like adapter protein Clone=701768	1.7600	1.2100	0.9900

The Lymphoma dataset is downloaded from Lymphoma/Leukemia Molecular Profiling Project (LLMPP) webpage [http://llmpp.nih.gov/lymphoma/data/figure1/figure1.cdt] as shown in Table 1. Human BCell contains about 4026 genes expressed in lymphoid cells or which are known as immunological or oncological importance with 96 conditions. There are three types of lymphomas such as diffuse large B-cell lymphoma (DLBCL), follicular lymphoma (FL), and chronic lymphocytic leukaemia (CLL) [1]. The entire data set includes the expression data of 4,026 genes each measured using a specialized cDNA microarray with its relevant Genbank accession number, Name and Clone IDs. A part of the dataset is chosen for the proposed work to classify lymphoma subtypes consists of hundred genes with gene expression values of 62 samples, with a total of 6200 samples and it is called as the Test dataset.

4.2.2 PREPROCESSING

Data pre-processing is an often neglected but important step in the data mining process.

Preprocessing is the process of removal of noisy data and filtering necessary information. The lymphoma dataset downloaded consist of noisy and inconsistent data. The multiple empty spots as shown in Table 2 are filled with values in the preprocessing phase.

GENE	NAME	VALUES	VALUES	VALUES	VALUES
GENE1835X	(Clone =1357915)	-0.1300		-0.2800	0.0400
GENE1836X	(Clone =1358277)	-0.3100	0.1600		0.2500
GENE1865X	(Clone =1358064)	-0.1200	0.5200		0.8300
GENE1933X	(Clone =1358190)	0.0500			0.2800
GENE1932X	(Clone =1336836)	-0.2600		-0.900	0.1500
GENE1931X	(Clone =1336983)	-0.5500			

4.2.3

REMOVAL OF NOISY DATA

The lymphoma dataset contains 4026 genes out of which certain gene expression values are missing. The missing data is imputed by knnimpute method. It replaces NaNs in data with the corresponding value from the nearest-neighbor column. The missing data in lymphoma dataset is replaced with nearest neighbor values as it is shown in Table.

The empty spots are filled with nearest values as data and the preprocessed values are given as input to the next process, called the ranking of genes.

4.2.4 RANKING OF GENES

Gene ranking simplifies gene expression tests to include only a very small number of genes rather than thousands of genes. The importance ranking of each gene is done using a feature ranking measure called T-Test which ranks the genes based on their statistical score. The t-test compares the actual difference between two means in relation to the variation in the data which is expressed as the

standard deviation of the difference between the means. T-Test includes the classes with different samples. The mean value of each gene expression in a class is calculated. In fact, the TS used here is a t-statistic between the centroid of a specific class and the overall centroid of all the classes. The T-Score of gene 'i' is defined as

$$Tsi = \max\left\{ \left| \frac{\bar{x}ik - \bar{x}i}{mksi} \right| k = 1,2..k \right\} \longrightarrow \quad \text{Eq.(1)}$$

Where there are K classes. Max (yk, k=1,2…k) is the maximum of all yk.

$$\bar{x}ik = \sum_{jsck} \bar{x}ij \Big/ nk \longrightarrow \quad \text{Eq.(2)}$$

Ck refers to class k that includes *nk* samples, *xij* is the expression value of gene i in sample j and *xik* is the mean expression value in class k for gene. *N* is total number of samples. *xi* is the general mean expression value for gene i. *si* is the pooled within-class standard deviation for gene i. The T-scores is calculated for the entire set of 4026 genes in Lymphoma dataset as shown in Table.

GENE ID	T-SCORE
GENE1943	0.2047
GENE880	0.1842
GENE324	0.1785
GENE1557	0.1641
GENE2231	0.1598
GENE289	0.1569
GENE1792	0.1559
GENE910	0.1548
GENE272	0.1547
GENE692	0.1541

4.2.5 FINDING INFORMATIVE GENES

Finding informative genes greatly reduces the computational burden and noise arising from irrelevant genes. The T-scores of the genes are sorted and the genes with the highest T-scores are ranked from 1 to 100. Hundred out of 4026 genes with the highest T-Scores are selected. Every gene is labeled after its

GENE	NAME	VALUES	VALUES	VALUES	VALUES
GENE1835X	(Clone =1357915)	-0.1300	-0.2800	-0.2800	0.0400
GENE1836X	(Clone =1358277)	-0.3100	0.1600	-0.2800	0.2500
GENE1865X	(Clone =1358064)	-0.1200	0.5200	0.1600	0.8300
GENE1933X	(Clone =1358190)	0.0500	0.0500	0.5200	0.2800
GENE1932X	(Clone =1336836)	-0.2600	-0.0900	-0.0900	0.1500
GENE1931X	(Clone =1336983)	-0.5500	-0.5500	-0.5500	-0.5500

importance rank. For example, Gene 1 means the gene ranked first as shown in Table. The genes with the highest scores are retained as informative genes.

GENE ID	T-SCORE	GENE
GENE1943	0.2047	1
GENE880	0.1842	2
GENE324	0.1785	3
GENE1557	0.1641	4
GENE2231	0.1598	5
GENE289	0.1569	6
GENE1792	0.1559	7
GENE910	0.1548	8
GENE272	0.1547	9
GENE692	0.1541	10

CONCLUSION

The fuzzy logic algorithm found a disproportionately large number of transcription factors in the roles of activators and repressors; however, not all of the activators and repressors found were transcription factors. Two possible reasons for this discrepancy are *1*) transcription factors are expressed at low levels and as such difficult to detect, and/or *2*) other gene products such as enzymes can indirectly regulate transcription. Transcription factors are generally present only at a very low concentration; thus changes in transcription factor expression levels can be difficult to detect using current expression profiling techniques. Presumably, if expression profiling technology were to become more sensitive, then the fuzzy logic algorithm would detect an even greater bias of transcription factors in the activator and repressor roles.

However, in many cases the expression level of a particular protein is not governed by the expression of a transcription factor, but instead by the concentration of some intracellular compound, such as Ca^{2+} concentration or cAMP levels, which in turn are controlled by enzymes inside the cell. In these cases, changes in the expression level of the enzyme have a "transcription-factor-like" effect and would be detected by the algorithm as an activator or repressor. From an drug design point of view, these "transcription-factor-like" enzymes are possibly more interesting than true transcription factors, because it is generally easier to change the activity of an enzyme in the cytosol with a drug than to block a true transcription factor in the nucleus. Moreover, the data set used in this study came from a single experiment in which cell cycle control was the main process of study.

Transcription factors that are not involved in pathways related to this cellular process might not show significant change in their expression and thus could not be evaluated by the fuzzy logic algorithm. To perform a more comprehensive survey on transcription factors, we are analyzing a data set that includes gene expression profiles of both wild-type and various mutant yeast cells. Many more transcription factors can be evaluated because the cellular processes they control have been perturbed.

An additional advantage to the fuzzy logic algorithm is that data can come from any source within an organism (tissue, cell type, treatment, or physiological state), and the output actually will be improved by deeper and more diverse data set. The reason for this improvement is that the algorithm needs to observe changes in the expression.

REFERENCES

1. Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ, and Davis RW. A genome-wide transcriptional analysis of the mitotic cell cycle. Mol Cell 2: 65–73, 1998.
2. Cox E. Fuzzy fundamentals. IEEE Spectrum 29: 58–61, 1992.
3. Deckert J, Perini R, Balasubramanian B, and Zitomer RS. Multiple elements and auto-repression regulate Rox1, a repressor of hypoxic genes in Saccharomyces cerevisiae. Genetics Soc Am 139: 1149–1158, 1995.

4. Fytlovich S, Gervais M, Agrimonti C, and Guiard B. Evidence for an interaction between the CYP1(HAP1) activator and a cellular factor during heme-dependent transcriptional regulation in the yeast Saccharomyces cerevisiae. EMBO J 12: 1209–1218, 1993.
5. Hach A, Hon T, and Zhang L. A new class of repression modules is critical for heme regulation of the yeast transcriptional activator Hap1. Mol Cell Biol 19: 4324–4333, 1999.
Abstract/FREE Full Text
6. Lodi T and Guiard B. Complex transcriptional regulation of the Saccharomyces cerevisiae CYB2 gene encoding cytochrome b_2: CYP1 (HAP1) activator binds to the CYB2 upstream activation site UAS1-B2. Mol Cell Biol 11: 3762–3772, 1991.
7. Prezant T, Pfeifer K, and Guarente L. Organization of the regulatory region of the yeast cyc7 gene: multiple factors are involved in regulation. Mol Cell Biol 7: 3252–3259, 1987.
8. Schneider JC and Guarente L. Regulation of the yeast CYT1 gene encoding cytochrome c_1 by HAP1 and HAP2/3/4. Mol Cell Biol 11: 4934–4942, 1991.
9. Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, and Church GM. Systematic determination of genetic network architecture. Nat Genet 22: 281–285, 1999.
10. Zadeh LA. Fuzzy logic and its application to approximate reasoning. Information Processing 74: 591–594, 1974.
11. Zhang L, Hach A, and Wang C. Molecular mechanism governing heme signaling in yeast: a higher-order complex mediates heme regulation of the transcriptional activator HAP1. Mol Cell Biol 18: 3819–3828, 1998.
12. Zitomer RS, Limbach MP, Rodriguez-Torres AM, Balasubramanian B, Deckert J, and Snow PM. Approaches to the study of Rox1 repression of the hypoxic genes in the yeast Saccharomyces cerevisiae. Methods Enzymol 11: 279–288, 1997.
13. Lingras, P. "Rough Set Clustering for Web Mining", Proceedings of 2002 IEEE International Conference on Fuzzy Systems. 2002.
14. Milligan G.W and Cooper M.C., "An examination of procedures for determining the number of clusters in a data set", Psychometrika, vol. 50, pp. 159-179, 1985.
15. Monmarche N. Slimane M, and Venturini G. Antclass, "Discovery of cluster in numeric data by an hybridization of an ant colony with the k-means algorithm", Technical Report 213, Ecole d' Ingenieurs en Informatique pour l'Industrie (E3i), Universite de Tours, Jan. 1999.
16. Selim S. Z and Ismail M. A, "K-Means type algorithms: a generalized convergence theorem and characterization of local optimality," in IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 6, No. 1, pp. 81--87, 1984.
17. Thangavel K and Ashok Kumar D, Department of Computer Science, "Simple Multi Pass Pattern Clustering Neural Networks", AIML Journal, Vol. 5, Issue (3), Dec., 2005.
18. Tou J.T. and Gonzalez R.C., "Pattern Recognition Principles", Massachusetts: Addision-Wesley, 1974.
19. Weiss SM and Indurkhya N. "Predictive Data Mining: a practical guide", Morgan Kaufmann, 1998.