UNSUPERVISED CLUSTERING OF GENE EXPRESSION PROFILES OF MICROARRAY DATA USING LIM AND PCA IN BIOINFORMATICS

[1]R.SHANMUGAVADIVU, [2] DR.N.NAGARAJAN

[1]Assistant professor PSG College of Arts& Science, Coimbatore, Tamilnadu,India
[2]Principal, Coimbatore Institute of Engineering and Technology, Coimbatore,Tamilnadu,India

[1]E-mail shan_vadivu@yahoo.com , [2]swekalnag@gmail.com

ABSTRACT

The development of microarray technology produces massive gene expression data sets. A major task for the experimentalist is to understand the structure in the huge data sets. Data generated by a scientific experiment always contain random noise. The situation is worst in the area of biology. Statistical methods must be used to accurately interpret large-scale experimental data. Microarray data is represented as a matrix with rows representing individual genes and columns representing conditions or experiments and very heterogeneous in nature. Because many proteins have unknown functions, and because many genes are active all the time in all kinds of cells, researchers usually use microarrays to make comparisons between similar cell types. As a result, we need to develop our ability to ‘‘see’’ the information in the massive tables of quantitative measurements that these approaches produce. This research addresses a suitable Micro array data clustering Algorithm which can be used to rearrange the gene expression profiles of microarray data for easy observation and knowledge discovery. In this research, a Lorenz Information Measure(LIM) based algorithm will be used to order the microarray data and after ordering the data, Principal Component Analysis(PCA) will be used to find the principal components in that order. Then the data will be clustered using a special kind of neural network called Self Organizing Maps (SOM).The Microarray data displayed after grouping will have some significance. After the clustering, we can see that the genes with similar expression patterns are grouped together under the related set of conditions. The Implementation of the proposed model will be done using Mat lab 6.5 under Windows operating system. The Performance of the system will be tested and evaluated with suitable gene expression data available for such kind of research.

Keywords: *Microarray, Lorenz Information Measure (LIM), Principal Component Analysis (PCA) , Self Organizing Maps (SOM)*

1 INTRODUCTION

1.1Datamining and Bioinformatics

Bioinformatics is the science of organizing and analyzing biological data. These data come from many different fields, including the studies of deoxyribonucleic acid (DNA), ribonucleic acid (RNA), genes, and proteins and how these molecules affect the functioning of the body such as the brain. In the last few years, many advances have been made in research and quite a bit of new data has been recorded. However, each new discovery produces more questions and more research topics. This has led to an explosion of data. “These fields allow for faster analysis of biological data and the discovery of many previously unknown biological trends.”

Genome database mining is the identification of the protein-encoding regions of a genome and the assignment of functions to these genes on the basis of sequence similarity homologies against other genes of known function. Gene expression database mining is the identification of intrinsic patterns and relationships in transcriptional expression data generated by large-scale gene expression experiments. Proteome database mining is the identification of intrinsic patterns and relationships in translational expression data generated by large-scale proteomics experiments. Improvements in genome, gene expression and proteome database mining algorithms will enable the prediction of protein function in the context of higher order processes such as the regulation of gene expression, metabolic pathways and signaling cascades.

1.2 About This Research

Microarray data is represented as a matrix with rows representing individual genes and columns representing conditions or experiments. Each cell value of the matrix is

$$\text{Log}\ \frac{(\text{expression of gene under specific condition})}{(\text{expression of gene under reference condition})}$$

Microarray data is very noisy and has a lot of missing values, which need to be dealt with before applying the proposed clustering. We adopted a very simple approach in which we just set the missing values to zero. However zero could be a valid cell value, indicating no change in the expression of the gene under a specific condition with respect to the test condition. Hence, a more appropriate approach would be to impute the missing values using an algorithm like k-nearest neighbors.

Microarray data can be visualized using three colors:

Black- represents a value of 0, indicating that the expression of the gene is unchanged in the specific condition as compared to the test condition.

Red- represents values > 0, indicating that the gene is over expressed in the specific condition as compared to the test condition.

Green- represents values < 0, indicating that the gene is under expressed in the specific condition as compared to the test condition.

A suitable Micro array data clustering Algorithm and a Pattern Classification algorithm will be used to order the gene expression profiles. The microarray data displayed after clustering will have some significance. The rows and columns have been reordered to group together rows and columns belonging to the same cluster. We can see that genes with similar expression patterns are grouped together under the related set of conditions. The proposed clustering algorithm is applicable to microarray data since only a small subset of the genes participate in a cellular process of interest that takes place only in a subset of the conditions

2. DNA MICROARRAY

A DNA microarray is a collection of microscopic DNA spots attached to a solid surface, such as glass, plastic or silicon chip forming an array. Scientists use DNA microarrays to measure the expression levels of large numbers of genes simultaneously. The affixed DNA segments are known as reporters, thousands of which can be used in a single DNA microarray. Microarray technology evolved from Southern Blotting, where fragmented DNA is attached to a substrate and then probed with a known gene or fragment. Measuring gene expression using microarrays is relevant to many areas of biology and medicine, such as studying treatments, disease and developmental stages. Microarrays can be fabricated using a variety of technologies, including printing with fine-pointed pins onto glass slides, photolithography using pre-made masks, photolithography using dynamic micro mirror devices, ink-jet printing, or electrochemistry on microelectrode arrays. The most common use of microarrays is to quantify mRNAs transcribed from different genes and which encode different proteins. RNA is extracted from many cells, ideally from a single cell type, then converted to cDNA or cRNA. The copies may be amplified by rtPCR. Fluorescent tags are enzymatically incorporated into the newly synthesized cDNA/cRNA or can be chemically attached to the new strands of DNA or RNA.

2.1 Microarray Analysis to Classify Genes and Phenotypes

Microarray experiments for simultaneously measuring RNA expression levels of thousands of genes are becoming widely used in genomic research. They have enormous promise in such areas as revealing function of genes in various cell populations, tumor classification, drug target identification. A major application of microarray technology is gene expression profiling to predict outcome in multiple tumor types. In a bioinformatics context, we can apply various data-mining methods to cancer datasets in order to identify class distinction genes and to classify tumors. A partial list of methods includes: (i) **Data preprocessing** (background elimination, identification of differentially expressed genes, and normalization); (ii) **Unsupervised clustering and visualization methods** (hierarchical, SOM, k-means, and SVD) (iii) **Supervised machine learning methods** for classification based on prior knowledge (discriminant analysis, support-vector machines, decision trees, neural networks, and k-nearest neighbors); and (iv) more ambitious **Genetic network** models (requiring large amounts of data) that are designed to discover biological pathways using such approaches as pair wise interactions, continuous or Boolean networks (based on a system of coupled differential equations) and probabilistic graph modeling based on **Bayesian networks**

3. CLASSIFICATION MODEL CONSTRUCTION

Model construction is building the model from the training set

- Each tuple/sample is assumed to belong a prefined class
- The class of a tuple/sample is determined by the class label attribute
- The training set of tuples/samples is used for model construction
- The model is represented as classification rules, decision trees or mathematical formulae

3.1 Model Usage

- Classify future or unknown objects
- Estimate accuracy of the model
- the known class of a test tuple/sample is compared with the result given by the mode
- accuracy rate = percentage of the tests tuples/samples correctly classified by the model

4. DATA CLASSIFICATION USING GENETIC ALGORITHMS

Genetic algorithms Optimization techniques that use processes such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution. Common image content properties are generally encoded as a series of histograms. Each histogram reflects a scale of an image property. Commonly used properties are: greyscale, color (red, green, blue or luminance, chroma and hue), line lengths, edge intensities, angle declinations, and texture. Lawrence Information Measure (LIM) is a histogram based technique which is commonly used to measure the information content of a typical image in content based image retrieval applications(CBIR).In this research, the novel idea of measuring the information of gene expression profiles of microarray data using LIM is proposed. The LIM profiles of the gene expression profiles of microarray data will be used to order the expression data to group inter-related items.

4.1 Histogram

The distribution of gray levels occurring in an image is called gray level histogram. It is a graph showing the frequency of occurrence of each gray level in the image versus the gray level itself. The plot of this function provides a global description of the appearance of the image.The histogram of a digital image with gray levels in the range [0,L-1] is a discrete function.

$P(r_k) = n_k / n$

where,

r_k is the Kth gray level

n_k is the number of pixels in the image with that gray level.

n is the total number of pixels in the image.

$K = 0,1,2,….,L-1.$

$L = 256.$

$P(r_k)$ gives an estimate of the probability of occurrence of gray level r_k .

4.2 The Lorenz Information Measure (LIM)

The Lorenz Information Measure (LIM) $(P_1,….,P_n)$ is defined to be the area under the Lorenz information curve . Thus from Figure the area of LIM C_a is greater than the area of LIM C_b. Clearly, 0 < = LIM $(P_{1,…….,} P_n)$ < = 0.5. For any probability vector $(P_{1,…….,}P_n)$, LIM $(P_{1,…….,}P_n)$ can be calculated by the first ordering P_i's, then calculating the area under the piecewise linear curve. Since LIM $(P_{1,…….,}P_n)$ (which can be expressed as the sum of $f(P_i)$, and $f(P_i)$) is a continuous convex function, LIM $(P_{1,…….,}P_n)$ is considered as an information measure.

Formula to compute LIM:

1) $P_0 = 0$
2) W = Width (Interval of histogram) has to be equal distribution
3) 0 <= LIM $(P_{1,…….,}P_n)$ <= 0.5
4) LIM = {W*P1/2+(W*P1+W*(P2-P1)/2)+,,,,,+(W*Pn-1+W*(Pn-Pn-1)/2)}

/ 2*# of pixels

$= \{W^* \sum^{n-1} P_i + W^* (\sum^n (P_i - P_{i-1})/2)\}$ / 2*# of pixels

$= W^* (\sum^{n-1} P_i + \sum^n (P_i - P_{i-1})/2)$ / 2*# of pixels

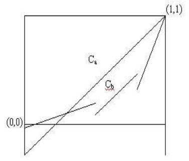


Figure 1 : Finding The LIM

Intuitively, the LIM can be regarded as a global content-based information measure. To compute the area of histograms the histogram intervals are sorted from low to high, and the resulting off-diagonal shape measured through differentiation.

4.3 SOM

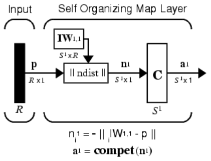
The Self-Organizing Maps (SOM) is a neural network model that is capable of projecting high dimensional input data onto a low-dimensional array. This nonlinear projection produces a two-dimensional "feature map" that can be useful in detecting and analyzing features in the input space. The SOM gives an intuitively appealing two-dimensional map of the multidimensional data set, and it has been successfully used for vector quantization and speech recognition. However, like its sequential counterpart, the SOM generates a suboptimal partition if the initial weights are not chosen properly. Further, its convergence is controlled by various parameters such as the learning rate and a neighborhood of the winning node in which learning takes place. It is possible that a particular input pattern can fire different output units at different iterations; this brings up the stability issue of learning systems. The system is said to be stable if no pattern in the training data changes its category after a finite number of learning iterations. This problem is closely associated with the problem of plasticity, which is the ability of the algorithm to adapt to new data. For stability, the learning rate should be decreased to zero as iterations progress and this affects the plasticity. ANNs use a fixed number of output nodes which limit the number of clusters that can be produced.

4.3.1 Principle of Self-Organizing Maps

In this proposed implementation of the SOM based algorithm, to minimize the time, there will be two major steps.

- In the first step we create a SOM (neural network) with ***x*** input neurons and ***k*** ourputs. (x is the total number of pixels in a block and ***k*** is the number of segments needed). The ***n*** vectors were prepared from the n non-overlapping blocks of the image. These non overlapping blocks will reflect all the textures in that image. The neural network will be self organized to map the ***k*** types of texture blocks. So this phase will require training.
- In the second step, the same neural network is used to map the N number of patterns (pattern vectors) into k number of segments. Since the neural network is already organized for k types of patterns, this phase will not require any training.

The architecture for this SOFM



The architecture for this SOFM is shown

Figure 2 : Self Organizing Feature Map

The learning rate and the neighborhood distance used to determine which neurons are in the winning neuron's neighborhood are altered during training through two phases.

Phase 1: Ordering Phase

This phase lasts for the given number of steps. The neighborhood distance starts as the maximum distance between two neurons, and decreases to the tuning neighborhood distance. The learning rate starts at the ordering-phase learning rate and decreases until it reaches the tuning-phase learning rate. As the neighborhood distance and learning rate decrease over this phase, the neurons of the network typically order themselves in the input space with the same topology in which they are ordered physically.

Phase 2: Tuning Phase

This phase lasts for the rest of training or adaption. The neighborhood distance stays at the tuning neighborhood distance, (which should include only close neighbors). The learning rate continues to decrease from the tuning phase learning rate, but very slowly. The small neighborhood and slowly decreasing learning rate fine tune the network, while keeping the ordering learned in the previous phase stable. The number of epochs for the tuning part of training (or time steps for adaption) should be much larger than the number of steps in the ordering phase, because the tuning phase usually takes much longer.

Competitive layers can be understood better when their weight vectors and input vectors are shown graphically. The diagram below shows 48 two-element input vectors represented as with `+' markers.

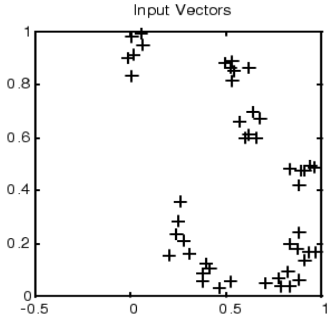


Figure 3: Clustering Using Self Organizing Feature Map

The input vectors above appear to fall into clusters. We can use a competitive network of eight neurons to classify the vectors into such clusters.

4.4 Principal Component Analysis

PCA assumes that all the variability in a process should be used in the analysis therefore it becomes difficult to distinguish the important variable from the less important.

4.4.1 Principal Components

A data set $\mathbf{x}_i$, $(i = 1,\ldots,n)$ is summarized as a linear combination of orthonormal vectors (called principal components):

$$f(\mathbf{x},\mathbf{V}) = \mathbf{u} + (\mathbf{x}\mathbf{V})\mathbf{V}^T$$

where $f(\mathbf{x},\mathbf{V})$ is a vector valued function, $\mathbf{u}$ is the mean of the data $\{\mathbf{x}_i\}$, and $\mathbf{V}$ is an $d \times m$ matrix with orthonormal columns. The mapping $\mathbf{z}_i = \mathbf{x}_i\mathbf{V}$ provides a low-dimensional projection of the vectors $\mathbf{x}_i$ if $m < d$.

The PCA estimates the projection matrix $\mathbf{V}$ minimizing

$$R_{emp}(\mathbf{x},\mathbf{V}) = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i - f(\mathbf{x}_i,\mathbf{V})\right\|^2$$

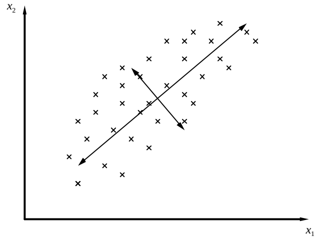


Figure 4: The first principal component

The first principal component is an axis in the direction of maximum variance.Consequently, Principle Component Analysis (PCA) replaces the original variables of a data set with a smaller number of uncorrelated variables called the *principle components*. If the original data set of dimension D contains highly correlated variables, then there is an effective dimensionality, d < D, that explains most of the data. The presence of only a few components of d makes it easier to label each dimension with an intuitive meaning. Furthermore, it is more efficient to operate on fewer variables in subsequent analysis. Using the built-in functions of Matlab we can do PCA in simple steps or even in one step with new versions of Matlab.

4.4.2 The Metric Used to Measure the Performance

In order to compare clustering results against external criteria, a measure of agreement is needed. Since we assume that each record is assigned to only one class in the external criterion and to only one cluster, measures of agreement between two partitions can be used.The Rand index or Rand measure is a commonly used technique for measure of such similarity between two data clusters.

Given a set of n objects S = {O1, ..., On} and two data clusters of S which we want to compare: X = {x1, ..., xR} and Y = {y1, ..., yS} where the different partitions of X and Y are disjoint and their union is equal to S; we can compute the following values

a is the number of elements in S that are in the same partition in X and in the same partition in Y,

b is the number of elements in S that are not in the same partition in X and not in the same partition in Y,

c is the number of elements in S that are in the same partition in X and not in the same partition in Y,

d is the number of elements in S that are not in the same partition in X but are in the same partition in Y.

Intuitively, one can think of a + b as the number of agreements between X and Y and c + d the number of disagreements between X and Y. The rand index, R, then becomes,

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

The rand index has a value between 0 and 1 with 0 indicating that the two data clusters do not agree on any pair of points and 1 indicating that the data clusters are exactly the same. In this project, a yeast microarray data named "yeastall_public.txt" collected from Internet will be used to test the algorithm.

5. THE BLOCK DIAGRAM SHOWING PROPOSED SYSTEM

The following diagram shows the overall design of the system

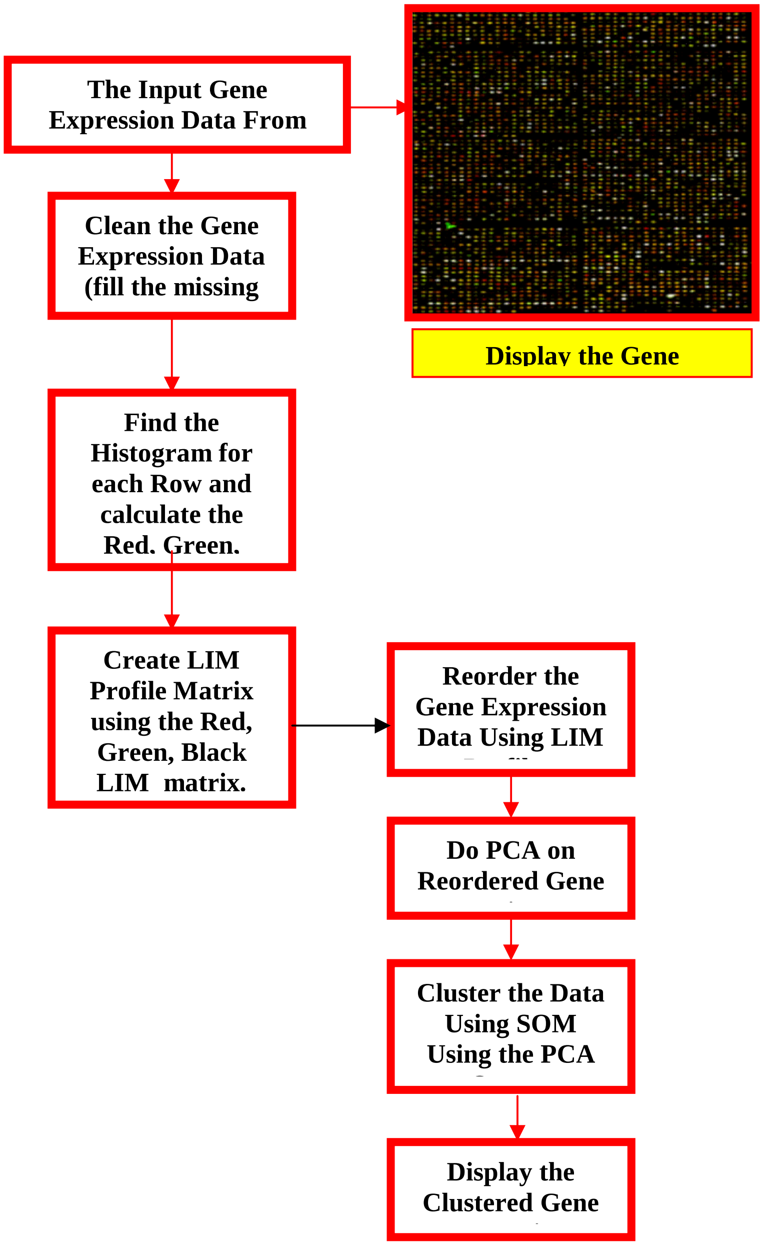
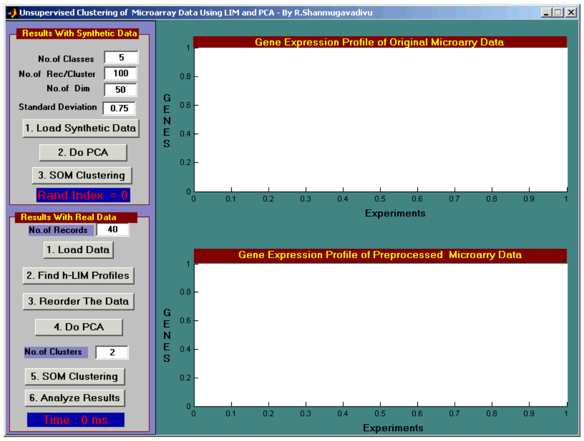

The Input Gene Expression Data From
Display the Gene
Clean the Gene Expression Data (fill the missing
Find the Histogram for each Row and calculate the Red. Green.
Create LIM Profile Matrix using the Red, Green, Black LIM matrix.
Reorder the Gene Expression Data Using LIM
Do PCA on Reordered Gene
Cluster the Data Using SOM Using the PCA
Display the Clustered Gene

Figure 5 : Block Diagram Showing Proposed System

6. THE MAIN INTERFACE CREATED FOR TESTING THE ALGORITHM

The following Matlab GUI interface was created to evaluate the performance of the micro array data clustering algorithm. The top axis control was used to plot the original gene expressions of microarray data. The bottom axis control was used to plot the sorted gene expressions of microarray data. The controls in the left side panel is used to test the algorithm with different parameters.



Unsupervised Clustering of Microarray Data Using LIM and PCA - By R.Shanmugavadivu

Results With Synthetic Data

No.of Classes: 5
No.of Rec/Cluster: 100
No.of Dim: 50
Standard Deviation: 0.75

Rand Index = 0

Results With Real Data

No.of Records: 40
No.of Clusters: 2

Time : 0 ms

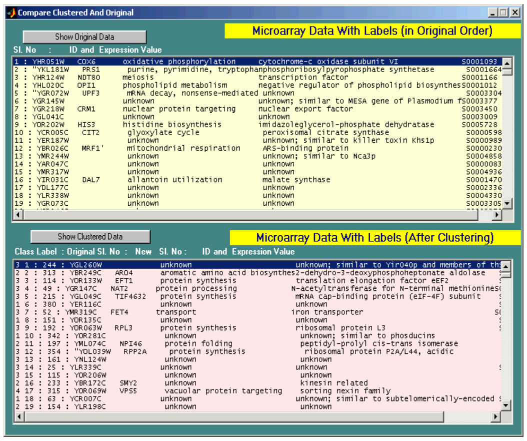
Figure 6 : The Main Interface

7.IMPLEMENTATION AND RESULTS

7.1Data set selection

We have searched for some available microarray experimental dataset through Internet, and found some interesting Yeast data sets. Yeast subjected to heat shock. This dataset consists of several heat shock experiments where expression was measured over time. The data is available in Text format ('Yeast_data.txt' seperated by TABs), or in Star office spreadsheet format.. The microarray data are provided in text/tab-delimited format. These files can be read with any text editor program. The recommended way to look at the data is with a spreadsheet program like MS Excel. Each column is identified by a header, the order of columns in the file can vary.

In the following screen shots, the descriptions/labels of the expression data in Original order (top) and rearranged Order(bottom). If we noted carefully, we can see that the items in the bottom list were grouped and signifies something important. In this case we see the records belonging to Protein Synthesis were got grouped after the microarray data clustering process.



Compare Clustered And Original

Microarray Data With Labels (in Original Order)

Sl. No : ID and Expression Value

```
1 : YHR051W   COX6     oxidative phosphorylation     cytochrome-c oxidase subunit VI              S0001093
2 : "YKL181W   PRS1     purine, pyrimidine, tryptophanphosphoribosylpyrophosphate synthetase       S0001664
3 : YHR124W   NDT80    meiosis                        transcription factor                         S0001166
4 : YHL020C   OPI1     phospholipid metabolism        negative regulator of phospholipid biosynthesS0001012
5 : "YGR072W   UPF3     mRNA decay, nonsense-mediated unknown                                      S0003304
6 : YGR145W            unknown                        unknown; similar to MESA gene of Plasmodium fS0003377
7 : YGR218W   CRM1     nuclear protein targeting      nuclear export factor                        S0003450
8 : YGL041C            unknown                        unknown                                      S0003009
9 : YOR202W   HIS3     histidine biosynthesis         imidazoleglycerol-phosphate dehydratase      S0005728
10 : YCR005C   CIT2     glyoxylate cycle               peroxisomal citrate synthase                 S0000598
11 : YER187W            unknown                        unknown; similar to killer toxin Khs1p       S0000989
12 : YBR026C   MRF1'    mitochondrial respiration      ARS-binding protein                          S0000230
13 : YMR244W            unknown                        unknown; similar to Nca3p                    S0004858
14 : YAR047C            unknown                        unknown                                      S0000083
15 : YMR317W            unknown                        unknown                                      S0004936
16 : YIR031C   DAL7     allantoin utilization          malate synthase                              S0001470
17 : YDL177C            unknown                        unknown                                      S0002336
18 : YLR338W            unknown                        unknown                                      S0004330
19 : YGR073C            unknown                        unknown                                      S0003305
```

Microarray Data With Labels (After Clustering)

Class Label : Original Sl. No : New Sl. No : ID and Expression Value

```
3 1 : 244 : YGL260W              unknown                        unknown; similar to Yir040p and members of th
2 2 : 313 : YBR249C   ARO4       aromatic amino acid biosynthes2-dehydro-3-deoxyphosphoheptonate aldolase   S
3 3 : 114 : YOR133W   EFT1       protein synthesis              translation elongation factor eEF2         S
3 4 : 49 : YGR147C   NAT2       protein processing             N-acetyltransferase for N-terminal methionineS
3 5 : 215 : YGL049C   TIF4632    protein synthesis              mRNA cap-binding protein (eIF-4F) subunit  S
1 6 : 380 : YER116C              unknown                        unknown                                    S
3 7 : 52 : YMR319C   FET4       transport                      iron transporter                           S
1 8 : 151 : YOR135C              unknown                        unknown                                    S
3 9 : 192 : YOR063W   RPL3       protein synthesis              ribosomal protein L3                       S
1 10 : 342 : YOR281C              unknown                        unknown; similar to phosducins
2 11 : 197 : YML074C   NPI46      protein folding                peptidyl-prolyl cis-trans isomerase
3 12 : 354 : "YOL039W   RPP2A      protein synthesis              ribosomal protein P2A/L44, acidic
3 13 : 161 : YNL124W              unknown                        unknown
3 14 : 25 : YLR339C              unknown                        unknown                                    S
3 15 : 115 : YOR206W              unknown                        unknown
2 16 : 233 : YBR172C   SMY2       unknown                        kinesin related
4 17 : 315 : YOR069W   VPS5       vacuolar protein targeting     sorting nexin family
1 18 : 63 : YCR007C              unknown                        unknown; similar to subtelomerically-encoded S
2 19 : 154 : YLR198C              unknown                        unknown
```

Figure 7 : The Diagram Showing Grouped Features

7.2The Performance Analysis

The following table presents the overall results of the time study made against the performance of the proposed gene microarray data micro array data clustering algorithm. These results were obtained with synthetic data sets where the ideal class labels were generally not available with real micro array data.

Number of Classes : 5
Number of Dimension of Data :50

Table 1 : The Performance with respect to different No. of Records

Sl.No	No.of Records	Time Taken for		Rand Index
		Doing PCA (Sec)	Clustering the Data (Sec)	
1	50	0.0310	1.3120	0.9012
2	100	0.0310	2.2040	0.9221
3	150	0.0470	3.0620	0.9668
4	200	0.0310	3.9380	0.9733
5	250	0.0630	4.8590	0.8973
6	300	0.0940	5.6870	0.9273
7	350	0.1250	6.6410	0.9477
8	400	0.2970	7.5000	0.9288
9	450	0.3750	8.3590	0.9168
10	500	0.5320	9.2500	0.9656

7.3The Performance in Terms of Speed

The following line chart shows the overall results of the time study made against the performance of the proposed gene microarray data micro array data clustering algorithm. The performance seems to be linear with respect to the increase of number of records in the microarray dataset. The time taken for taking LIM is insignificant when comparing it with the total time taken to complete the operation. So the time taken for creating LIM profile is not taken in to account.

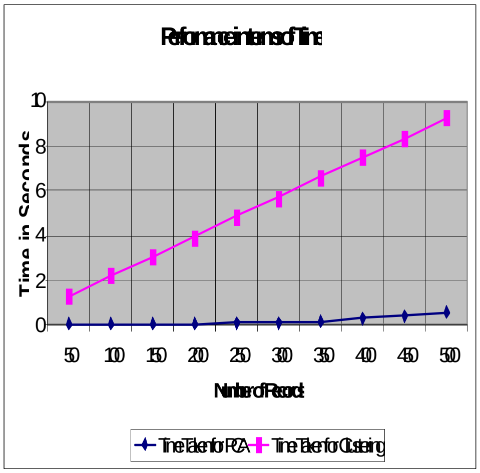


Figure 9 : The Performance in terms of accuracy

7.4 The Performance in Terms of Accuracy

The Performance in terms of classification accuracy is almost constant (between 0.9 to 0.98) if the number of records is below 1000.

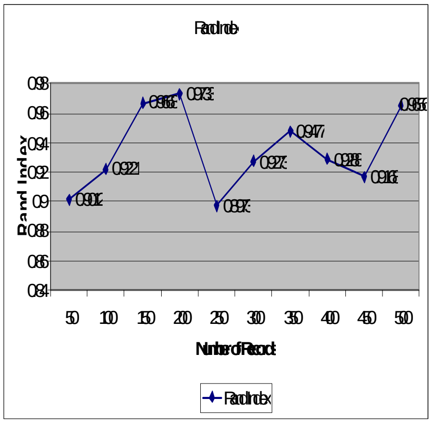


Figure 8 : The Performance in terms of Time

8. CONCLUSION

The proposed Microarray Analysis System was successfully implemented using different techniques such as LIM, PCA, and SOM using Matlab 6.5 which has no readymade functions related with bioinformatics. The gene expression data was successfully clustered using the system.The system was tested with real data of yeast gene expression profiles which is collected from internet. The experiments were repeated with different number of input records. The results of the proposed algorithm were significant and has qualities for further investigation. From the results we observe that the genes with similar functions are grouped together. This can be used to determine the biological function of unknown genes by comparing it with the nearby known genes. So the system can be used to analyze the complex results of microarray experiments. It will act as a interactive tool for microarray data analysis.As shown in the tables and charts in the results section, the performance seems to be linear with respect to the increase of number of records in the microarray dataset. The Performance in terms of classification accuracy is almost constant (between 0.9 to 0.98) if the number of records is below 1000.

In this research, LIM profile of the gene expression and the SOM based clustering were used to reorder the microarray for better understanding of the data. It was observed that the classification accuracy as well as the speed of the algorithm degrades with the in crease of total records and the dimension of the data. Future works may address the possibilities of improving accuracy in the cases of very large data sets. Other data mining techniques can also be to used to discover more knowledge from the sorted microarray data. All theses issues can be addressed in future works.

REFERENCES

[1] Michael B.Eisen, Paul T. Spellman, Patrick O.Brown, and David Botstein, “Clustering analysis and display of genome-wide expression patterns”. 1998.

[2] Erica Kolatch, “ Clustering Algorithms for Spatial Databases A survey”. 2001.

[3] Eunice Yin, Xiaomeng Wu, “ Clustering on Microarray Dataset (CMD)”.2002.

[4] Bezdek, J.C, “Pattern Recognition with Fuzzy Objective Function Algorithms”. Plenum Press, New York. 1981

[5] Ron Shamir. “Algorithms in Molecular Biology Lecture Notes”. Tel Aivi University. 2000

[6] Timothy Slidel. “Distributed Computing in the Life Science.” 1998.

[7] Arun K Pujari , “ Data mining techniques” Universities Press – 2001

[8] Michael J. A. Berry, Gordon S. Linoff – “Mastering Data Mining” – Wiley Computer Publishing 2001

[9] Pieter Adriaans, Dolf Zantinge - “Data Mining” – Addison Wesley Longman 1999

[10] Rhonda Delmater, Monte Hancock –“ Data Mining Explained” Butterworth-Heinemann 2001