Two-Way Clustering Analysis using Parallel Fuzzy Approach for Microarray Gene Expression Data

Dwitiya Tyagi-Tiwari
Department of Mathematics & Computer Applications
Maulana Azad National Institute of Technology,
Bhopal, India

Sujoy Das
Department of Mathematics & Computer Applications
Maulana Azad National Institute of Technology,
Bhopal, India

Namita Srivastava
Department of Mathematics & Computer Applications
Maulana Azad National Institute of Technology,
Bhopal, India

ABSTRACT
A microarray measures the expression levels of thousands of genes at the same time. Clustering helps to analyze microarray gene expression data. The characteristic of gene expression data is its coherent structure with regards to genes and samples. In this paper we have implemented a biclustering algorithm to identify subgroups of data which shows correlated behavior under specific experimental conditions. In the process of finding biclusters, Fuzzy C-means clustering is used to cluster the genes and samples with maximum membership function. Dimensionality and reducing the gene shaving is done using principal component analysis & gene filtering with the function respectively. This method obtains highly correlated sub matrices of the gene expression dataset. It is also observed that it identifies important co-regulated genes and samples at the same time. Principal component analysis is also verified the concatenation of small biclusters into bigger one. Biclustering is a NP-hard problem [10] therefore we have implemented biclustering in multi-core parallel environment to reduce the computational time of the algorithm. Data level and task level parallelism is used to develop the algorithm on MATLAB Parallel computing toolbox with multicore platform. We have compared the results with other parallel & sequential algorithm to show the effectiveness of the algorithm.

Keywords
Microarray, gene expression, Multicore platform, Biclustering, MATLAB parallel computing, PCA, gene entropy.

1. INTRODUCTION
Microarray helps in studying the variations of many genes simultaneously. With the development of microarray techniques, a lot of work has been done on the analysis of gene expression data. Microarray experiments, identifies co-expressed genes that share similar expression patterns [1]. Clustering is the key initial step in the analysis of gene expression data to find the co-expressed genes. In the previous studies it has been found that the genes showing similar expression patterns are likely to be involved in same cellular processes. When the pattern that related genes showing similar transcriptional behavior under a subset of condition is called bicluster [3,4]. It is a type of subspace clustering. Generally, a gene only belongs to one cluster but in practicality, a gene involves many cellular processes, so a gene may belong to more than one cluster. A better solution would be to introduce fuzzy concepts at the time of gene clustering because impact of biclustering will be more obvious. Strong correlations of expression patterns between the genes indicated as co-regulation [1, 2] and are controlled by same regulatory mechanisms. One of the major characteristic of gene expression data is to find meaningful clusters or groups with respect to both genes and samples dimensions.

Hartigan [22] developed the two way clustering approach for biclustering, and Cheng and Church [7] were the first to use this concept for microarray data analysis. Wei et al. [9] proposed a parallel biclustering algorithm based on anti-monotones property of the quality of the data sets with their sizes. Tewfik [11] et al. proposed parallel biclustering of genes with coherent evolutions. It finds all biclusters with a specified minimum numbers of genes and conditions in the datasets. Jiang et al [15] proposed mining approach for coherent gene clusters from microarray gene expression dataset, they have presented two approaches namely sample-gene search and gene-sample search to mine a set of coherent gene clusters. The serial version of Interrelated Two-Way Clustering which was proposed by [5] is parallelized in [18], and is implemented and tested to find biclusters. Chandra et.al [6] have also used this concept to find biclusters in gene expression data with fuzzy approach. In this paper we have used the concept of fuzzy c-means to cluster the genes and sample dimensions.In the next step maximize the sum of distances between the genes having maximum membership function to find the gene centers.For the preparation of biclusterswe have used gene entropy filtering and Principal component analysis for the gene shaving process.The main goal of the algorithm is to find coherent values of gene bicluster one set at a time with this new approach. Parallel approach of the algorithm on multicore processor is provide the better performance than sequential one.

2. MATERIAL AND METHODS
2.1 Proposed Algorithm
Tang et.al [5] proposed an interrelated two-way clustering algorithm to identify important genes and samples simultaneously. The same algorithm is used in MATLAB multicore environment [18]. Classical K-means approach is used to form clusters of genes and samples.

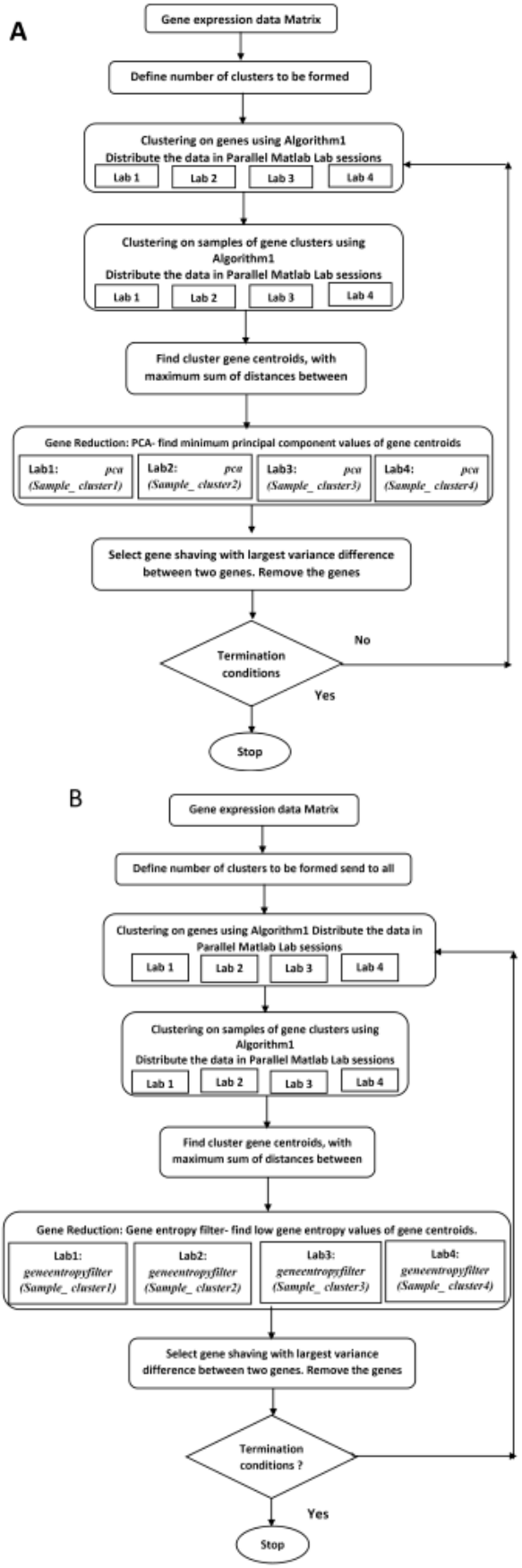

Fig.1: Schematic diagram of Interrelated Two-Way clustering with Parallel Fuzzy C-means- (A) PCA (B) Gene entropy filter

Yeast data: In this paper we have used cell cycle of the budding yeast S. cerevisiae made by Cho et al. (1998).This data set contains the expression profiles of 2945 genes made by Tavazoie et al. (1999). In that selection, the data for the time points 90 and 100 min are excluded.

In the procedure of finding biclusters from the high dimensional gene expression data, it is not possible to directly identify the sample patterns or significant genes. This kind of processing is used to find the relationships of sample clusters and gene groups to make partial or approximate patterns. Then use this pattern directly to eliminate the irrelevant genes [5, 6]. Fig. 1 (A) and (B) shows the algorithm for finding biclusters.

In schematic diagram we have shown the steps of the sequential algorithm which includes with start of the algorithm fuzzy c-means algorithm to find gene clusters and sample clusters of the dataset. Next step is to find gene centroids with maximization of the sum of distances between the samples which is having maximum membership values in the clusters. Gene reduction is based on two methods first is gene dimension reduction based on PCA and second is gene filtering based on gene entropy calculation of the gene centroids. We set gene shaving point for eliminating the genes from the groups either PCA or gene entropy filtering.

Then remove the gene centroids below that gene shaving point. Fig.3 and Fig.4 which is showing the computational Time.

Termination criteria is number of iteration perform, based on number of biclusters we want to search. Here we set number of biclusters are 30 or number of iterations are 50.

2.2 Two Way Clustering using Fuzzy approach

In the proposed algorithm parallel fuzzy clustering is performed on both genes and the samples simultaneously. Genes and samples clustered using parallel Fuzzy C-means clustering method. Fuzzy c-means is a data clustering technique where in each data point belongs to a cluster with some degree that is specified by a membership grade. Fuzzy clustering algorithm links each gene to the clusters via a real-valued vector which lies between 0 and 1. The gene which index value close to 1 indicates a strong association to clusters and the index value close to 0 indicate the absence corresponding cluster. The computation is done iteratively. The original algorithm is based on minimization of the following objective functions to compute further calculation according to Bezdek 1981 [5]:

$$J(K, m) = \sum_{k=1}^{K} \sum_{i=0}^{N} (u_{ki})^{m} d^2(x_i, c_k) \quad (1)$$

$$d^2(x_i, c_k) = (x_i - c_k)^T A_k (x_i - c_k) \quad (2)$$

$$\text{with } \sum_{k=1}^{K} u_{ki} = 1; 0 < \sum_{i=0}^{N} u_{ki} < N \quad (3)$$

$$\text{where } 1 \leq i \leq N \text{ and } 1 \leq k \leq K.$$

In Equation (1), K is number of cluster and N is number of data objects, m is constant real-valued number which controls the 'fuzziness' which is greater than 1 (choose from (2)), u_{ki} is the degree of membership of data object x_i in cluster k, and $d^2(x_i, c_k)$ is the square distance from data object x_i to cluster centroid c_k.

From Equation (1) cluster centroid c_k and membership vectors u_{ki}, these parameters obtained from the following equations by Bezdek 1981[5]:

Step1: Initialize inputs K, m choose any product norm metric for calculation of *d2(***x***i* , **c***k)*. Select randomly *K* samples as initial centroids **c***(0) k* and then form partitions of all others samples around these centroids to obtain the initial partition matrix *U(0)* = [*uki*], *k* = 1, . . . , *K* and *i* = 1, . . . , *N*. At step *l*, *l* = 1, 2, . . . , perform the following steps:

Step2: Compute the matrix of centroids $\mathbf{c}_k^{(l)}$:

$$c_k^{(l)} = \frac{\sum_{i=1}^{N}\left(u_{ki}^{(l-1)}\right)^m x_i}{\sum_{i=1}^{N}\left(u_{ki}^{(l-1)}\right)^m}; \text{ k=1,2-------K} \quad (4)$$

Step3: Computation of membership values $u_{ki}^{(l)}$:

$$u_{ki}^{(l)} = \begin{cases} \dfrac{1}{\sum_{s=1}^{K}\left[\dfrac{d^2\left(x_i c_k^{(l)}\right)}{d^2\left(x_i c_s^{(l)}\right)}\right]^{\frac{2}{(m-1)}}} \\ 0 \\ \dfrac{1}{|I_i|} \end{cases} \quad (5)$$

If $I_i = \emptyset. \quad \forall_i \in \tilde{I}_i \, , \; \forall_i \in$

After the computation of the distance matrix between data samples and cluster centers, this fuzzy partition matrix function computes the membership values for the data set. This process makes the highest membership values for the cohesive clusters formation.

Step4: Repeat step (2) and (3).

After several passes from step (2) to (3), the algorithm will stop and after that we have to build a fuzzy partition matrix U whose numbers are smaller than a specified level [3, 7].
Using Fuzzy C-means algorithm in genes and sample dimension perform the following steps to find the biclusters. In Step1 and 2 we have used parallel Fuzzy C-means approach to cluster the dataset. Parallel fuzzy c-means clustering is based on data level parallelism, in which the dataset divided equally among the labs and send the initial cluster centers to all the labs. Calculations of all the labs are shown in Fig2 Algorithm1.

To perform this parallel FCM we have used MATLAB parallel computing toolbox with quad core processor.

We have taken the number of clusters as an input for gene and samples, here we take K=2. In the proposed algorithm genes and samples are clustered using parallel fuzzy c-means algorithm as shown in Algorithm1 Fig.2. The schematic diagram of proposed method is illustrated in Fig. 1 (A) using PCA for gene dimension reduction, and (B) using Gene entropy filter for gene dimension reduction.

Matlab client:
- ✓ Calculate the cluster centroids ($c_i = c_k^{(l)}$) by equation 4.
- ✓ Broadcast it to all the workers. So that all workers have same cluster centroids at the beginning of FCM looping.
- ✓ Collect the results-

Matlab workers:
- ✓ Calculate membership matrix ($u_{ki}^{(l)}$) with equation 5.
- ✓ Compute the distances as in equation 2 from each data points in local dataset to all clusters centers; update the partition matrix.
- ✓ Calculate new cluster centers as in equation 4.
- ✓ If difference between previous and new cluster centers $\leq \varepsilon$ or iteration is 100 then stop; otherwise, continue.
- ✓ Final results send to the client.

Fig 2: Algorithm1 - Parallel Fuzzy C-means

Following steps of interrelated two-way clustering analysis using parallel fuzzy approach is performed as follows to find the biclusters:

Step1: Clustering on genes: Parallel fuzzy c-means algorithm (Fig.2-Algorithm1) is used to cluster the genes into *K* groups, denoted by G1 and G2. Find K genes with maximum membership function.

Step2: Clustering on Samples: Similarly parallel fuzzy c-means algorithm (Fig.2-Algorithm1) is applied on gene groups to cluster the samples into *K* clusters.

Step3: Find gene centers: maximize the sum of distances between the genes having maximum membership function. We find gene centers of K sample clusters, denoted as $C_j (1 \leq j \leq 2^K)$.

Step4: Sort and reduce the genes: apply gene entropy filter **or** PCA on gene centers found in Step3 for gene reduction from sample clusters.

This process is parallelized to distribute the gene centers to MATLAB Lab sessions to calculate the PCA or gene entropy filter for gene reduction. If we take K=2 then the clustered data is distributed in two MATLAB Lab sessions, and when we take K=4 then the gene centers distributed in four MATLAB Lab sessions.

Gene reduction: here we have used two gene reduction methods to compute the biclusters and compared them. First is principal component analysis, and second is gene entropy filter. Both methods are applied in separate with each other. The definition of these techniques are as follows:

Principal-component analysis

Principal-component analysis is a useful approach that can be used for dimensionality reduction and feature extraction. It is a linear projection method that determines a new dimensional space, which captures the maximum information present in the original matrix. PCA has been used with the direct MATLAB function *pca (Gene_Expression_Data).*

Entropy

Entropy is a function of correlation which provides the amount of information that may be gained by an observation of a system and it measures variation or changes in a series of

events. Entropy denotes the diversity of information in given data set which makes it suitable for clustering genes. In this paper we calculate low gene entropy of given microarray dataset. For the measurement of interdependency of two random genes X and Y we have used a MATLAB function *geneentropyfilter(Data, Names, 'Percentile', PercentileValue).*

If K=2 then sample cluster distributed in two MATLAB Lab sessions, and if K = 4 then the distribution of the sample clusters in four MATLAB Lab sessions:

Table 1: Calculations of PCA in Labs

Lab_index 1	Lab_index 2	Lab_index3	Lab_index4
pca (Sample_cluster1)	pca (Sample_cluster2)	pca (Sample_cluster3)	pca (Sample_cluster4)

OR

Table 2: Calculation of Gene Entropy in Labs

Lab_index 1	Lab_index 2	Lab_index3	Lab_index4
geneentropyfilter(Sample_cluster1, Names, 'Percentile', PercentileValue)	geneentropyfilter(Sample_cluster2, Names, 'Percentile', PercentileValue)	geneentropyfilter(Sample_cluster3, Names, 'Percentile', PercentileValue)	geneentropyfilter(Sample_cluster4, Names, 'Percentile', PercentileValue)

Gene shaving

Gene shaving is the important part to find good biclusters in gene expression dataset. Here we have used two methods for principal component analysis and gene entropy filter approach to reduce the genes based on sample classification. In this algorithm gene shaving point is not fixed, it is based on according to the method to be used for gene dimension reduction. For PCA we calculate the principal components for each gene centroids and remove the genes below the points which shows largest variance among two genes, and for gene entropy filter we calculate the gene entropy values for gene centroids and remove gene centroids with low gene entropy values below the points and remaining genes are send for the next iteration.

Step5: Termination condition: Repeat the process go to Step1 until termination condition is satisfied. Stop the iteration if the iteration reached 50; otherwise continue. If the number of genes dropped is less than 150 Or number of biclusters reached 30, the iterations are stopped.

3. RESULTS AND DISCUSSION

Parallel version is implemented on yeast Saccharomyces cerevisiae cell cycle expression data from [12, 21]. We demonstrate that feature filtering to outperform the gene entropy filter and PCA based method. There are cases where the analysis, based on a small set of selected features, outperforms the best score reported when all information was used. Our method calls for an optimal size of the relevant feature set.

The algorithm is implemented on MATLAB version 2010a with parallel computing toolbox on Intel(R) Core 2 Duo CPU 750 @ 2.67GHzprocessor, 16GB RAM. This version can open 12 parallel lab sessions at a time.

We evaluate performances of the proposed algorithm with both the methods of gene dimension reduction.

3.1 Performance evaluation with PCA:

We applied the proposed algorithm in yeast S. Cerevisiae used by Cho et al [21], and yeast S. Cerevisiae Tavazoie et al. [12]. Our goal is to analyze the effect of PCA as gene dimension reduction method in mentioned data sets. We have calculated principal components for the gene centroids to reduce the genes to get good biclusters. Principal components are uncorrelated and ordered such that the nth largest variance among all principal components. We calculate the principal components for each gene samples group and select the gene shaving point with maximum variance among the gene centroids, remove the genes below the shaving points. Repeat the process with remaining genes until termination criteria is met.

3.2 Performance evaluation with gene entropy filter

According to the literature entropy should below for order configuration of the data and high for disorderly configurations of the data sets.

As the clustering results, the entropy should be low for every close within the clusters. With the help of entropy the property we generate the rank the list of gene centroids. Remove the gene centroids which is having greater values of gene expression data as it is least important. We reduce the genes based on the gene entropy ranked list. The property of entropy is used to rank the informativeness of the genes. We have calculated the gene entropy filter for gene centroids found in step 3 and choose the gene shaving point based on the largest variance among the two given genes, and remove the genes below that point.

Performance of the algorithm show only serial and parallel fuzzy c-means approach, and another is to show serial and parallel fuzzy interrelated two way clustering on two-dimensional gene expression dataset (i) yeast S. Cerevisiae used by Cho et al [21], and yeast S. Cerevisiae Tavazoie et al. [12] with two or four clusters, run concurrently on two to six lab sessions. The computational speed of parallel fuzzy c-means as compared to serial fuzzy c-means and is given in Table 1. Table 1 shows Turnaround Time of forming clusters which is obtained in Step1 and Step2 of the proposed algorithm. Table 2 shows computation time of the proposed two-way fuzzy c-means algorithm which is shown in Fig.3 and Fig. 4. In Table1 and Table 2 k is set to 2 clusters and four clusters, TSeq refers to Turnaround Time of sequential run, TPar refers to Turnaround Time of parallel algorithm, and TDiff refers to time difference between the serial and parallel runs of the algorithms. The formula of time difference equation 6 between the algorithms is as follows:

$$TDiff = TSeq - TPar \quad (6)$$

The speedup ratio is calculated according to the following equation 7:

$$Speedup = \frac{TSeq}{TPar} \quad (7)$$

Linear speedup or ideal speedup is obtained when speedup is equal to number of lab sessions. In our case. For the performance evaluation we have used different values of k = 2 and 4. As tables shown the parallel version of the algorithms performs better than sequential run of the algorithm.

Table 3: The Turnaround Time of serial fuzzy c-means versus parallel fuzzy c-means, for k = 2 and 4 for gene and sample clustering. k is number of clusters.

Dataset	k	TSeq	TPar	TDiff	Speedup
2945	2	11.119	10.101	0.018	1.02
	4	21.043	19.981	0.062	1.03
6149	2	21.910	19.821	1.089	1.6
	4	32.992	23.879	1.113	1.4

Similarly we have calculated the computational speed of serial fuzzy interrelated two-way clustering versus parallel fuzzy interrelated two-way clustering given in Table 2 where k is set to 2 clusters and 4.

Table 4: The Turnaround Time of serial fuzzy two-way clustering versus parallel fuzzy two-way clustering for k=2 and 4. k is number of clusters.

Dataset	k	TSeq	TPar	TDiff	Speedup
2945	2	190.891	140.802	50.089	1.36
	4	310.29	300.09	10.2	1.03
6149	2	551.021	394.222	156.799	1.398
	4	712.091	598.991	113.1	1.19

Turnaround Time of the algorithm is calculated in seconds. From Table 1 and Table 2, it is clear that parallel version of algorithm is faster than sequential algorithm. The comparison graph of Turnaround Time between sequential and parallel fuzzy two way clustering analysis is shown in Fig. 3 with principal component calculations. Computation time of parallel algorithms is calculated in 6 parallel MATLAB sessions for k = 2 clusters.

Interrelated two way clustering is originally developed by [5], they have applied the algorithm on various gene expression datasets Multiple sclerosis (MS) and leukemia patient samples. Chandra et al. [6] applied interrelated two-way approach on the colon cancer data set and the leukemia data set.

We have calculated this time of work for PCA gene dimension reduction method to eliminate the gene centroids. The computation time of the gene centroids elimination is based on gene filtering is not showing higher variation of bicluster computation.

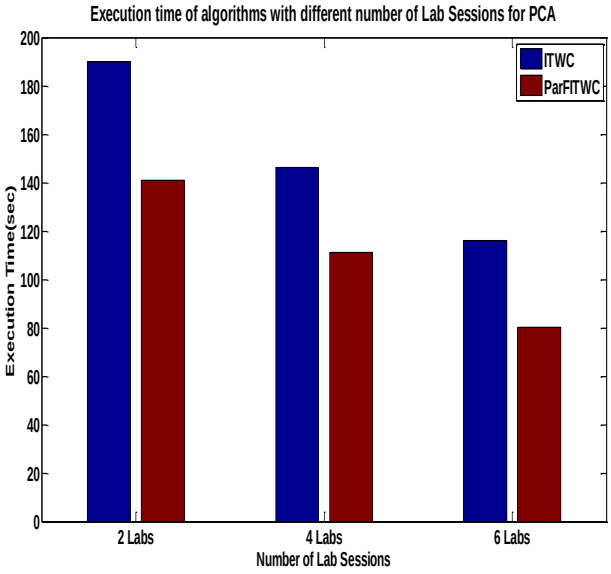

Fig. 3: Parallel algorithm computation time with different lab sessions with PCA

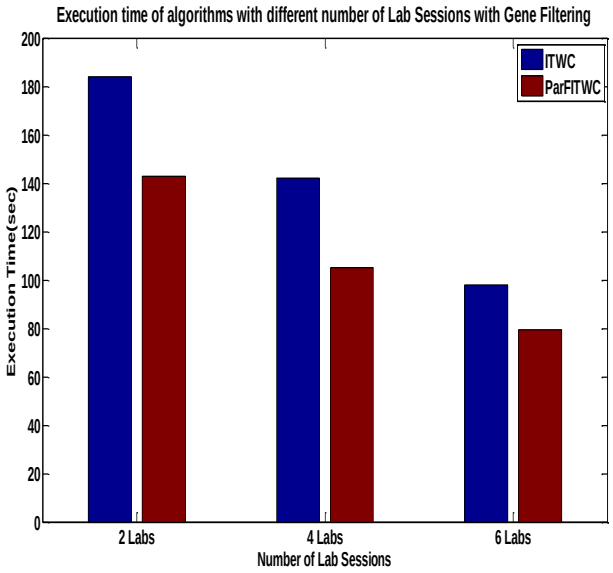

Fig. 4: Parallel algorithm computation time with different lab sessions with Gene filtering.

Table3 and Table 4 is showing the comparative analysis of gene dimension reduction process with gene entropy filtering and PCA calculations. In the Table we have given common biclusters found with gene filtering and PCA calculation approach with number of genes and samples in the biclusters. When k = 2, 6 common biclusters from Cho et al [21], and 4 common biclusters from yeast S. Cerevisiae by Tavazoie et al [12], and k = 4, total 8 common biclusters found for Cho et al.[21] dataset and 5 common biclusters from Tavazoie et al. [12] dataset out of 30 biclusters. Table 3 and 4 have given the description of the common biclusters found with number of genes and samples respectively for both datasets.

Common biclusters of gene dimension reduction techniques of gene entropy filter and PCA for number of clusters k = 2 of both datasets used in this study.

Table 5: Common biclusters found by gene entropy filter and PCA for the datasets for k = 2

Dataset	Biclusters number	Number of Genes	Number of Samples
Tavazoie et al	11	882	4
	12	718	3
	17	512	4
	22	293	6
Cho et al	9	1021	3
	10	989	4
	16	999	3
	21	821	4
	24	822	2
	29	629	3

Common biclusters of gene dimension reduction techniques of gene entropy filter and PCA for number of clusters k = 4 of both datasets used in this study.

Table 6: Common biclusters found by gene entropy filter and PCA for the datasets for k = 4

Dataset	Biclusters number	Number of Genes	Number of Samples
Tavazoie et al	12	700	3
	17	512	3
	20	310	5
	22	233	4
	27	230	2
Cho et al	11	913	4
	13	819	3
	15	710	5
	18	515	3
	21	444	4
	22	421	5
	29	312	5
	30	218	3

As a result of study the gene entropy filter process is comparatively batter than principal component in unsupervised approach. According to the literature [13][14][16] principal component analysis is computationally higher. Entropy filtering have given most top ranked biclusters as compared to principal component in our case. During the calculation we have observed that the Turnaround Time for these gene dimension reduction techniques, there were not showing very high variations between them as shown in Fig. 3 and 4.

4. CONCLUSION

In this paper, another new approach of two way clustering of gene expression data has been proposed in multicore environment. The approach uses fuzzy C-means clustering algorithm for grouping genes and sample dimensions. The experiments on two gene expression used by yeast S. cerevisiae Cho et al. [21] and Tavazoie et al.[12] data sets show that biclustering of two gene expression data. Gene Entropy filtering shows the good performance in gene shaving. After applying the different filtering techniques, the best among them is Gene Entropy Filtering since it removed the maximum waste and noisy data. Gene entropy used to rank the informative genes and reduction of lower entropy gives least important feature or genes. Performance of PCA shows that it is a good powerful bicluster verification tool as it have the ability to remove correlation of the data. The study of results of biclustering also shows that the number of clusters of genes and samples are also affects, we have taken number of clusters two and four respectively. For cluster size two gives larger biclusters whereas cluster number 4 have given smaller one as compared to two for both the dataset. It can be assumed that large number of clustering with smaller biclusters shows more cohesiveness of the biclusters. The overall results demonstrated that this approach outrun parallel fuzzy interrelated two way clustering even with reduced number of genes.

5. ACKNOWLEDGEMENTS

One of the authors Dwitiya Tyagi-Tiwari would like to thank Charmi Panchal for helping me to analyze the algorithm and Vladimir Rogojin for the helpful discussion and support to implement the concept.

6. REFERENCES

[1] Daxin Jiang; Chun Tang; Aidong Zhang, "Cluster analysis for gene expression data: a survey," Knowledge and Data Engineering, IEEE Transactions on, vol.16, no.11, 1370-1386, Nov. 2004.

[2] Daxin Jiang, Jian Pei and Aidong Zhang, "GPX: Interactive Mining of Gene Expression Data", 30th VLDB Conference, Toronto, Canada, 2004.

[3] G. Kerr, H.J. Ruskin, M. Crane and P. Doolan, “Techniques for Clustering Gene Expression Data”, Computers in Biology and Medicine 38, 2008, 283-293.

[4] Erfaneh Naghieh and Yonghong Peng, "Microarray Gene Expression Data Mining: Clustering Analysis Review", Aug 20, 2009.

[5] Chun Tang and Aidong Zhang, “Interrelated Two-Way Clustering and Its Application on Gene Expression Data ", International Journal on Artificial Intelligence Tools, 2005; Vol. 14, No. 4; 577-598.

[6] B. Chandra, S. Shankera, Saroj Mishra, "A new approach: Interrelated two-way clustering of gene

expression data", Statistical Methodology 3, 2006, 93–102.

[7] Y. Cheng and G.M. Church, "Biclustering of Expression Data", in Proc. Of American Association for Artificial Intelligence, 2000.

[8] Y. Kluger, R. Basri, J.T. Chang, M. Gerstein, Spectral biclustering of microarray data: coclustering genes and conditions, Genome Res. 13 (4), 2003 703–716.

[9] Liu Wei And Chen Ling, "A Parallel Algorithm For Gene Expressing Data Biclustering", Journal Of Computers, Vol. 3, No. 10, October 2008, 71-77.

[10] Sara C. Madeira and Arlindo L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey", IEEE/Acm Transactions on Computational Biology and Bioinformatics Vol 1, No. 1, January-March 2004, 24-45.

[11] A.H. Tewfik, A.B. Techagang and I. Vertatsehitsch, "Parallel Identification of Gene Biclusters with Coherent Evolutions", IEEE Transaction on Signal Processing, Vol. 54, No. 6, June-2006.

[12] Tavazoie,S., Hughes,J.D., Campbell,M.J., Cho,R.J. and Church,G.M. (1999) Systematic determination of genetic network architecture. Nat. Genet., 22, 281–285.

[13] Zhang Yanjie, Veronique Prinet and Wu Shuanhu, "A Principal Component Analysis Based Microarray Data Bi-Clustering Method", 2nd International Conference on Biomedical Engineering and Informatics, October 2009.

[14] K. Y. Yeung and W. L. Ruzzo, "Principal Component Analysis for Clustering Gene Expression Data", Bioinformatics Vol. 17 no. 9 2001, 763-774.

[15] Genevera I, Allen and Mirjana Maletic-Savatic, "Sparse non-negative generalized PCA with applications to metabolomics", Bioinformatics, Vol. 27 no. 21 2011, 3029-3035.

[16] Asa Ben-Hur and Isabelle Guyon, "Detecting Stable Clusters Using Principal Component Analysis", In Functional Genomics: Methods and Protocols. M.J. Brownstein and A. Kohodursky (eds.) Humana press, 2003, 159-182.

[17] Christoph Bartenhagen, Hans-Ulrich Klein, Christian Rckert, Xiaoyi Jiang and Martin Dugas, "Comparative study of unsupervised dimension reduction techniques for the visualization of microarray gene expression data", BMC Bioinformatics 2010.

[18] Dwitiya Tyagi, Sujoy Das, and Namita Srivastava, Parallel Two-way Clustering for Microarray Gene expression data', International Journal of Computer Science Trends and Technology, Vol. 3 Issue 3, May-June 2015.

[19] Wei Shen, Guixia Liu, Ming Zheng, Zhangxu Li, Yi Zhong, Jianan Wu, Chunguang Zhou, A Novel Biclustering Algorithm and Its Application in Gene Expression Profles, Journal of Information & Computational Science 9: 11 (2012), 3113–3122.

[20] Bezdec, J.C., Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[21] Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ, Davis RW, 'A genome-wide transcriptional analysis of the mitotic cell cycle', Molecular Cell, Vol. 2, July, 1998, 65–73.

[22] Hartigan J.: "Direct Clustering of a Data Matrix", J Am Stat Assoc, 67(337), pp. 123-129, 1972.