Enhanced Hierarchical Clustering for Gene Expression data

Geetha.T
Department of Computer Applications
National Institute of Technology
Trichirappalli, Tamil Nadu, India

Michael Arock
Department of Computer Applications
National Institute of Technology
Trichirappalli, Tamil Nadu, India

ABSTRACT

Micro arrays are used to assess the transcriptome of many biological systems that has generated an enormous amount of data. Cluster analysis is a technique used to group and analyze micro array data. Identification of groups of genes that manifest similar expression patterns is a key step in the analysis of gene expression data. Hierarchical clustering is the one of the clustering techniques used for this purpose. In this paper, we design an enhanced hierarchical clustering algorithm which scans the dataset and calculates distance matrix only once unlike other papers, (up to authors' knowledge). Our main contribution is to reduce time, even when a large database is analyzed. Also, the results of hierarchical clustering are represented as a binary tree which gives clarity in grouping and further helps to find clustered objects easily. Our algorithm is able to retrieve number of clusters with the help of cut distance and measures the quality with validation index in order to obtain the best one; does not require initial parameter like number of clusters.

Categories and Subject Descriptors

H.3.3 [Information System**s**]: Information search and Retrieval – *Clustering*.

General Terms

Algorithms, Design, Measurement.

Keywords

Micro array, Hierarchical clustering, Gene expression data, Binary Tree

1. INTRODUCTION

Clustering of gene expression data helps to understand gene functions and regulations network and assists in the diagnostics of disease conditions [13] and effects of medical treatment. Clustering is a process of partitioning a dataset into separate groups ("Clusters") containing similar data items based on any similarity measure. It is an unsupervised classification technique; that does not require a priori knowledge of the groups which data members belong to. Objects within a cluster have high similarity in comparison to one another but very dissimilar with objects in other clusters [18].

Clustering algorithm can generally be classified as either Hierarchical or Partitional. The K-means algorithm [11] is a popular partitioning technique which groups objects into *K* groups of similar means. But it requires number of groups to be clustered prior to analysis; it is difficult for all types of databases. For categorical data a fuzzy K-modes [17] clustering algorithm had been used. The K-Sets [22] clustering algorithm was implemented for categorical and mixed data. The K-sets algorithm uses connected component instead of means/modes and achieves more stable and better results. However, K-sets cannot deal with outliers or with large databases. Self Organizing Map (SOM) that represents the high dimensional data as low dimensional data. Self-organizing map was used by Tamayo et al. [32] to identify clusters in the yeast cell cycle data set and human hematopoietic differentiation data set.

The SOM [20] had been applied to group gene expression data. However, it is difficult to find clear boundaries from results of SOM. This problem is removed by combining [1] SOM with K-means clustering. A variant of hierarchical clustering algorithms has been proposed by Eisen et al. [12] to identify groups of co-expressed yeast genes and fibroblast genes. Parallelization [10, 7] of hierarchical clustering approach has been implemented for cancerous disease and to handle very large databases. Hierarchical clustering [23] of gene expression data has been implemented by Luo and Khan. SLINK: [31] an optimally efficient algorithm had been developed for the single link clustering method. Efficient agglomerative hierarchical clustering [8], complete link methods [9] are also used for grouping. Fast agglomerative [16] hierarchical clustering algorithm using locality-sensitive hashing has been used for clustering. CHAMELEON: [19] hierarchical clustering using dynamic modeling, groups objects together.

Alon et al. [2] developed a two-way clustering technique to detect groups of correlated genes and tissues. Chang and Church proposed Bi-clustering [5] of gene expression data. Later Coupled two-way clustering [14] analysis has been implemented for gene microarray data. The OPTOC [30], one prototype take one cluster method is used for clustering. Correlation based clustering algorithm [33] has used external validation index but it requires more memory and execution time. Enhanced Correlation search technique [29] has been applied for clustering gene expression data. CLUSTAG [3], hierarchical clustering and graph based method has been applied for selecting tag SNPs.

Recently, many clustering algorithms based on evolutionary computing such as genetic algorithms have been introduced, and only a couple of applications opted for particle swarm optimization [27]. Particle swarm optimization (PSO), a population-based algorithm (Kennedy & Eberhart) [21] simulates bird flocking or fish schooling behavior to build a self-evolving system. A hybrid technique [34] combining the K-means algorithm, Nelder–Mead simplex search, and particle swarm optimization, called K–NM–PSO, is proposed. The K-Means with PSO [6] has been proposed by Chen and Ye faced the problem of initial parameter requirement and local optima. Genetic

algorithms [26] typically start with some candidate solutions to the optimization problem and these candidates evolve towards a better solution through selection, crossover and mutation. Hybrid K-Means and Genetic algorithm [4] is implemented for data clustering. Large-scale analysis of seven different clustering methods and four proximity measures for the analysis of 35 cancer gene expression data sets are performed by Marcilio et al [24]. His results reveal that the finite mixture of Gaussians, followed closely by *k*-means, exhibited the best performance. An Ant-based clustering [25] system for knowledge discovery in DNA chip analysis is implemented by Minsoo Lee et al.

Duran et al. [11] analyzed various clustering algorithms revealed none of the algorithm is best suited for all the databases. Raja et al. [28] designed a metric to measure the effectiveness of clustering algorithms of genes by computing inter-cluster cohesiveness and intra-cluster separation with respect to biological features such as biological processes. Hierarchical clustering performs clustering in an easy way compare to other advanced complex algorithms and produced good results also.

The rest of the paper is organized as follows. Section 2 gives a gene representation in clustering and distance measure used in similarity calculation. Section 3 describes Hierarchical clustering algorithm and quality measure. Section 4 deals with dataset descriptions and results obtained from the algorithm. Section 5 presents conclusions and future work of the paper.

2. GENE REPRESENTATION

A Micro array experiment evaluates a large number of DNA sequences consisting of genes, cDNA clones or Expressed Sequence Tags under different conditions. These conditions may be a time based or tissue samples based. In this paper, Yeast cell data set with different time points are clustered. A gene expression data set from a micro-array experiment can be represented by a real-valued expression matrix. In this matrix, rows represent expression pattern of genes, columns represent expression profile of samples or experimental conditions.

Datasets are represented as set of genes G= {g1, g2, g3… gn}, where gi represents ith gene in the data set. The wij represents expression profile of ith gene at jth condition. Dataset with n genes and has m-dimensional vector of real numbers represented as follows.   Sample S

$$
\text{Gene G} \quad
\begin{matrix}
W_{11} & W_{12} & W_{13} & \ldots\ldots & W_{1m} \\
W_{21} & W_{22} & W_{23} & \ldots\ldots & W_{2m} \\
.. & .. & .. & \ldots\ldots & .. \\
.. & .. & .. & \ldots\ldots & .. \\
W_{n1} & W_{n2} & W_{n3} & \ldots\ldots & W_{nm}
\end{matrix}
$$

2.1 Similarity Metric

In cluster analysis, objects are grouped based on its similarity. Proximity measurement measures the similarity or distance between two objects. Here, similarities between two genes are measured using Euclidean distance. However, for gene expression data, the overall shapes of the gene expression profiles are of greater interest than the individual magnitudes of each feature. Similarities between two objects are calculated as follows:

$$D(g_1, g_2) = \sqrt{\sum_{j=1}^{m} (W_{1j} - W_{2j})^2} \qquad (1)$$

Here, g_1 and g_2 are two genes in the data set and m denotes dimension number or conditions. W_{1j} and W_{2j} are jth expression profile/feature values of g1 and g2 genes. Distance between pair of gene objects is calculated using formula (1) and distance matrix is represented as Δ.

$$\Delta =
\begin{matrix}
0 & D(g_1,g_2) & D(g_1,g_3) & \ldots & D(g_1,g_n) \\
D(g_2,g_1) & 0 & D(g_2,g_3) & \ldots & D(g_2,g_n) \\
D(g_3,g_1) & D(g_3,g_2) & 0 & \ldots & D(g_3,g_n) \\
.. & .. & .. & \ldots & .. \\
D(g_n,g_1) & D(g_n,g_2) & D(g_n,g_3) & \ldots & D(g_n,g_m)
\end{matrix} \quad (2)$$

Similarity measure is used to find closer objects. The objects with minimum distances are grouped together. So intra-cluster gap is minimal and inter-cluster gap is maximal.

3. HIERARCHICAL CLUSTERING

Cluster Analysis, also called data segmentation, has a variety of goals. All relate to grouping or segmenting a collection of objects into subsets or "clusters", such that those within each cluster are more closely related to one another than objects assigned to different clusters. Goal of cluster analysis is the notion of degree of similarity (or dissimilarity) between the individual objects being clustered. There are two major methods of clustering - hierarchical clustering and k-means clustering.

The hierarchical clustering method works by grouping data objects into a tree of clusters. In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to n clusters each containing a single object. Hierarchical Clustering is subdivided into agglomerative (Bottom-up/Merging) methods, which proceed by series of fusions of the n objects into groups, and divisive (Top-down/Splitting) methods, which separate *n* objects successively into finer groupings [5].

Hierarchical clustering may be represented by a two dimensional diagram known as dendrogram in Figure 1 which illustrates the merging made at each successive stage of analysis. The quality of pure hierarchical clustering method suffers from its inability to perform adjustment, once a merge or split decision has been executed. Recent studies have emphasized the integration of hierarica1 agglomeration with iterative relocation methods [15].
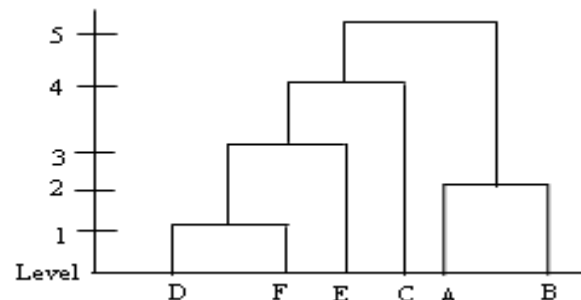

Figure 1. Dendrogram with 6 objects (Agglomerative)

3.1 Agglomerative Method

The agglomerative hierarchical clustering procedure produces a series of partitions of the data, Pn, Pn-1,.. P1. The first Pn consists of n single-object 'clusters', the last P1, consists of single group containing all n cases. At each particular stage the method joins together the pair of clusters which is the closest among other pair. This bottom-up strategy starts by placing each object in its own cluster. For example if we have five objects, initial number of clusters is also five and then merges atomic clusters into larger and larger clusters according to its similarity until certain termination conditions are satisfied. Similarity/Distance is calculated using Euclidean or Pearson method and stores it in a distance matrix. Here, lower triangular matrix values only are stored and accessed which reduces the amount of memory that also represented in one dimensional array.

Four different linkage measures are used to find similar objects with the help of distance matrix. The measures are Single linkage, Complete Linkage, Average Distance and Mean distance is represented in Figure 2. In this paper Single linkage clustering algorithm and complex linkage is used.
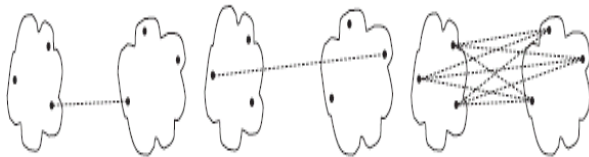
MIN(Single Link)   MAX(Complete Link)   Group average

Figure 2. Linkage Mesures.

3.2 Single Linkage Clustering

One of the simplest agglomerative hierarchical clustering methods is single linkage, also known as the nearest neighbor technique. The defining feature of the method is that distance between groups is defined as the distance between the closest pair of objects, where only pairs consisting of one object from each group are considered. In the single linkage method, D(r, s) is computed as   D(r, s) = Min {d (i , j) : Where object *i* is in cluster r and object *j* is cluster s }

Here the distance between every possible object pair (i, j) is computed, where object i is in cluster r and object j is in cluster s. The minimum value of these distances is said to be the distance between clusters r and s. In other words, the distance between two clusters is given by the value of the shortest link between the clusters. At each stage of hierarchical clustering, the clusters r and s , for which D(r, s) is minimum, are merged. Data points are treated as nodes of a graph with edges forming path between the nodes in a cluster. The merging of two clusters, *Ci* and *Cj* corresponds to adding an edge between the nearest pair of nodes in *Ci* and *Cj*, the resulting graph will generate a tree.

Distance matrix is calculated once in the first iteration. For the remaining iterations previous iteration distance matrix is used. It avoids the scanning of databases every time and thereby saves time. For the new distance matrix the distance between each objects are copied as it is from the existing one except newly merged clusters. Newly merged row and column values are assigned by finding the minimum distance between each object with newly clustered objects. For example, D and F are newly

merged clusters Distance matrix for DF with object A is calculated by taking min (DA, FA), this procedure repeated for all the objects. This avoids unnecessary calculations. The dimension of the cluster is reduced by one in successive iterations because of merging operation. Next iteration is followed with the new distance matrix calculated very recently. Then, minimum distance of all elements of the distance matrix is found. This process is repeated until all objects are grouped together.

Enhanced Hierarchical Clustering Algorithm

1. Assign each object as individual cluster like c1, c2, c3, .. cn where n is the no. of objects
2. Find the distance matrix D, using any similarity measure
3. Find the closest pair of clusters in the current clustering, say pair (r), (s), according to d(r, s) = min d(i, j) { i, is an object in cluster r and j in cluster s}
4. Merge clusters (r) and (s) into a single cluster to form a merged cluster. Store merged objects with its corresponding distance in Cophenetic Matrix. (Refer to subsection 3.3)
5. Update distance matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s). Adding a new row and column corresponding to the merged cluster(r, s) and old luster (k) is defined in this way:
   d[(k), (r, s)] = min d[(k),(r)], d[(k),(s)].
   For other rows and columns copy the corresponding data from existing distance matrix.
6. If all objects are in one cluster, stop. Otherwise, go to step 3.

In the Complete linkage method, D(r, s) is computed as   D(r, s) = Max { d(i, j) : Where object i is in cluster r and object j is cluster s }

In this paper, the results of clustering are represented as binary tree. In each iteration the details of the merged objects are represented a tree, with objects being represented as leaf node and the distance between them as non-leaf node. Each iteration's result forms one level up in binary tree. When the cut distance is given the binary tree will give number of clustered objects and members of each cluster.

The process is as follows: for a given cut distance, search operation starts from the root and searches next levels. If it finds the distance in the non leaf node at any level then it removes that non leaf node and gives its left and right children objects. Then this process repeated with its parent until it reaches root. Finally, it arranges the groups and gives the cluster number. Instead of using dendrogram, here the results are stored and retrieved from binary tree.

3.3 Quality Measure

We used Cross Correlation Coefficient (CCC) or Cophenetic Correlation Coefficient (CPCC) measure for hierarchical clustering. To compute the Cophenetic Correlation Coefficient of hierarchical clustering requires the information of Distance matrix and Cophenetic distance Matrix. Distance matrix is calculated in first iteration of hierarchical algorithm using Euclidean measure. The distance matrix is symmetric and so it needs only the lower triangular values.

The Cophenetic distance between two objects is the proximity at which an agglomerative hierarchical clustering technique puts the objects into the same cluster for the first time. For example when

merge cluster D and F into cluster (D, F) at distance 0.50, we Fill 4th row and 6th column and 6th row and 4th column with 0.5. In a Cophenetic distance matrix, the entries are the distances between each pair of objects. This process is repeated for all iterations.

Cophenetic Correlation Coefficient (CPCC): This coefficient calculates the correlation between these two distance matrices. One of the common uses of this measure is to evaluate which type of hierarchical clustering is best. It shows the goodness/fit of the clustering. The CPCC between two distance matrices X and Y are represented as r(X,Y) which is defined in Formula (3).

$$r(X,Y) = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{(N\sum X^2 - \sum X)^2)(N\sum Y^2 - (\sum Y)^2)}} \qquad (3)$$

The r(X,Y) yields a value between 0 and 1. The higher the correlation grows, the larger r(X,Y) gets. In particular, when X is identical to Y, r(X,Y) = 1. From the viewpoint of statistics, when $r(X,Y) \geq 0.7$, X is said to have a high correlation to Y.

4. DATA SETS and RESULTS

In this paper, we first deal with sample dataset and real life dataset. The sample data set has 3 attributes and 6 objects. Refer Table 1.

Table 1. Sample data set

X	Y	Object
1	1	A
1.5	1.5	B
5	5	C
3	4	D
4	4	E
3	3.5	F

The iterations for generating Binary Tree using single linkage are as follows:
1. Cluster D and cluster F into (D,F) at distance 0.5
2. Merge cluster A and cluster B into (A, B) at 0.71
3. Merge cluster E and (D, F) into ((D, F), E) at 1.00
4. Merge cluster ((D, F), E) and C into (((D, F), E), C) at distance 1.41
5. Merge cluster (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50

The iterations for generating Binary Tree using complete linkage are as follows:

1. Cluster A and Cluster C into (A, C) at distance 5.6. AC distance is 5.6
2. BE distance is 3.5
3. ACF distance is 2.5
4. BED distance is 1.0
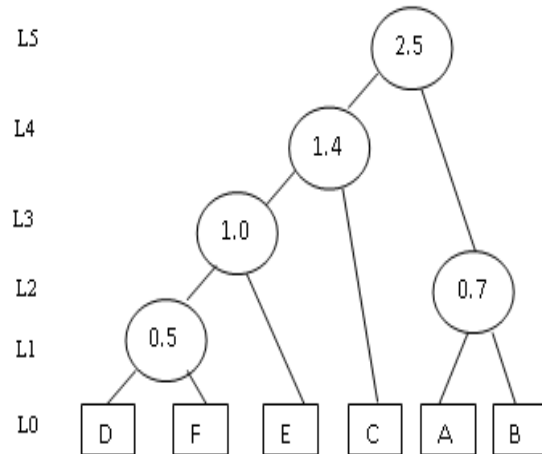5. ACFBED distance is 0.5

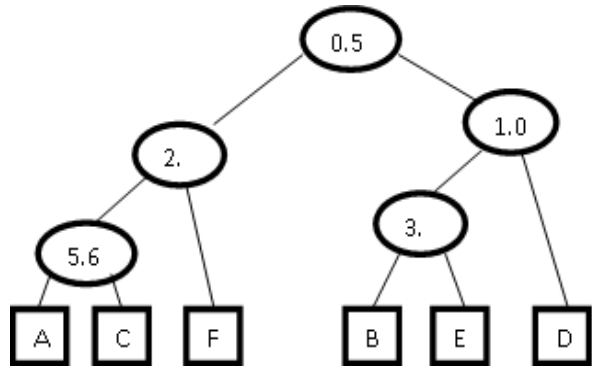Figure 3. Binary Tree Representation of Single link method

Figure 4. Binary Tree Representation of Complete link method

After clustering, the results of clustered objects and its merged distance are represented as binary tree in Figure 3 for single link, Figure 4 for complete link. The following process applied for single link algorithm. If the cut distance is 1.2 then it is between level 3 and level 4 shown in Figure 6. Cut at level 4 gives its right and left children as DFE and C (Figure 6) and it then proceeds with upper level (parent). Here parents happen be the root node (Figure 7). We cut the parent also and obtain its right side object (AB). The final clustered objects are (DFE) , (C) and (AB).
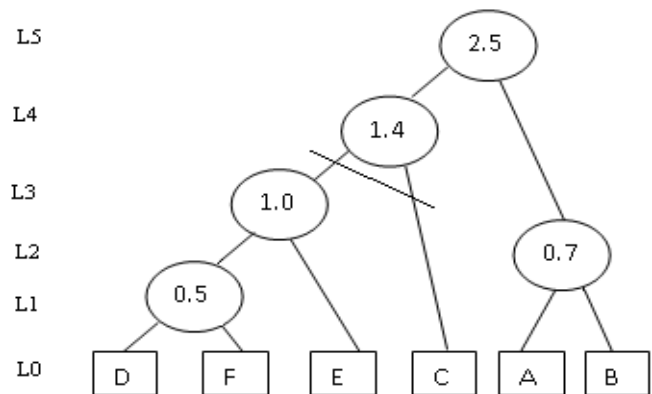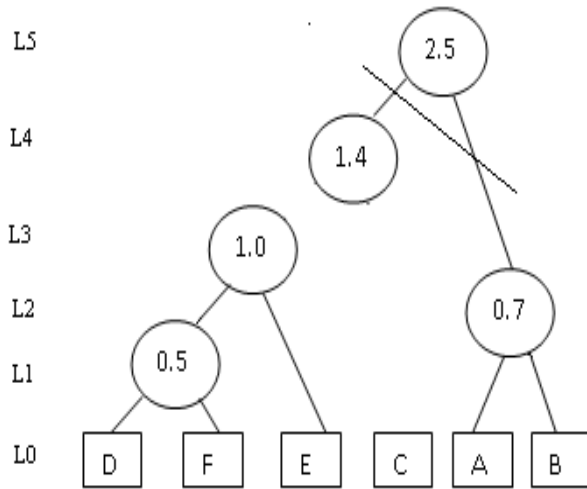

Figure 5. Binary Tree with cut at level 4
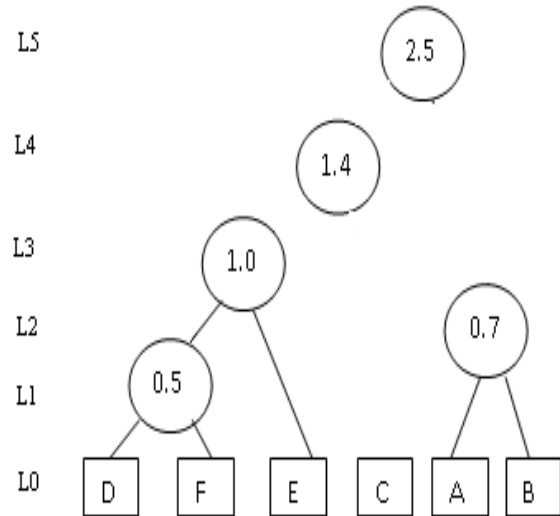
Figure 6. Binary Tree with cut at level 5

Figure 7. Left & Right objects cut at level 5 (Parent/Root)

The obtained distance matrix and cophenetic distance matrix are represented in Table 2 and Table 3. In this paper, distance matrix calculated only once (i.e in first iteration). Cophenetic distance matrix get values at the end of each iteration. For both matrices we need to fill only the lower triangular values. Complete matrix representation is given here.

Table 2. Distance Matrix

0	0.71	5.66	3.61	4.24	3.2
0.71	0	4.95	2.92	3.54	2.5
5.66	4.95	0	2.24	1.41	2.5
3.61	2.92	2.24	0	1	0.5
4.24	3.54	1.41	1	0	1.12
3.2	2.5	2.5	0.5	1.12	0

Table 3. Cophenetic Distance Matrix

0	0.71	2.5	2.5	2.5	2.5
0.71	0	2.5	2.5	2.5	2.5
2.5	2.5	0	1.41	1.41	1.41
2.5	2.5	1.41	0	1	0.5
2.5	2.5	1.41	1	0	1
2.5	2.5	1.41	0.5	1	0

Cophenetic Correlation Coefficient is simply correlation coefficient between distance matrix and Cophenetic matrix, Correl (Dist, CPCC). In our example Correl (Dist, CPCC)= 86.39%. As the value of the Cophenetic Correlation Coefficient is close to 100%, that the clustering is good. Refer Figure 8.

Table 4. Linkage Measures Vs Cophenentic Correlation Coefficient

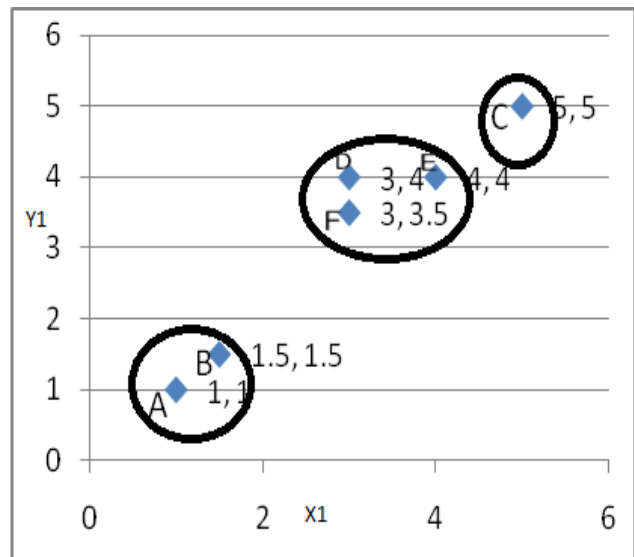| Techniques	CPCC
Single Link	0.863
Complete Link	0.539

Figure 8. Resultant Cluster objects

In our example we take yeast cell dataset with eight different attributes are clustered into ten groups. Figure 9 and figure 10, represents some of the yeast genes in cluster MIT (mitochondrial) and NUC (Nuclear). The elements we obtained in each groups are similar with the real grouping. Figure 10 shows four similarly expressed genes (AAR2_Yeast, BAF1_Yeast, ACE1_Yeast, ACE1_Yeast) of the yeast dataset.
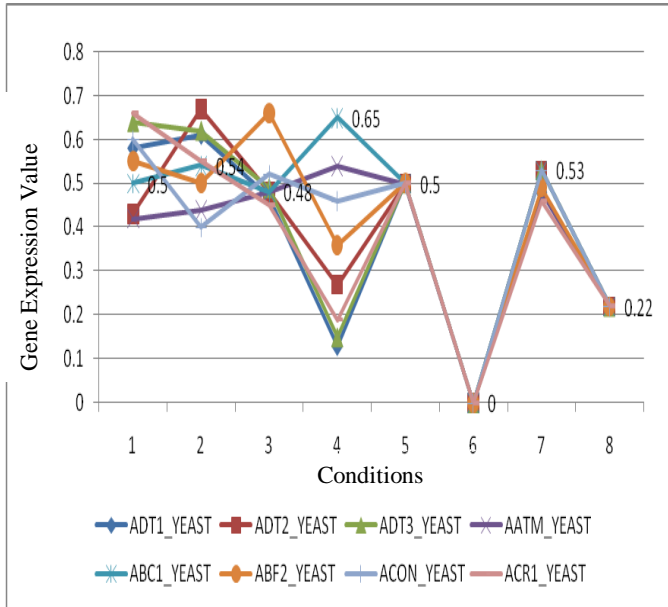
Figure 9. Yeast Cells in cluster MIT (mitochondrial)

Our enhanced hierarchical clustering is applied for breast cancer dataset with 9 features and 2 groups. The results obtained for this data set are also closer.
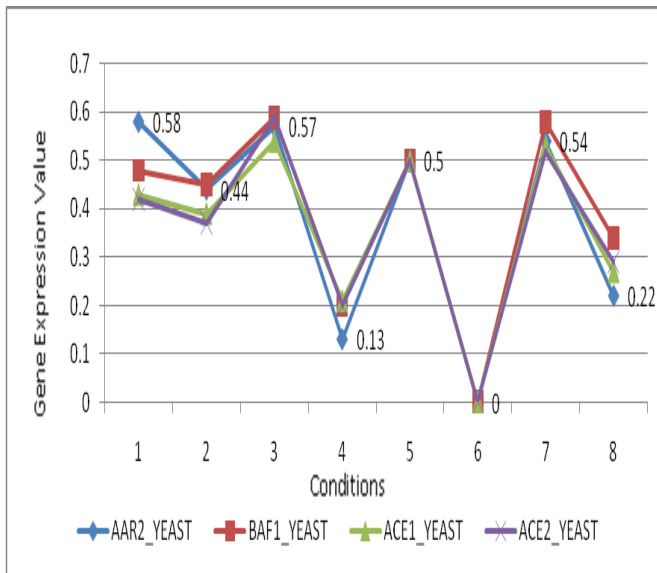

Figure 10. Yeast Cells in cluster NUC (nuclear)

5. CONCLUSIONS

This paper presents enhanced hierarchical clustering algorithm for gene expression data sets. In the previous papers, database scanning and distance matrix calculation are needed for all iterations. Our method reads the database and finds distance matrix only once, which reduces the amount of time. Also, our method requires the minimum space, as the lower triangular distance matrix can be represented in single dimensional array,

even when large databases are used. And, we represent the cluster results as a binary tree which gives clear grouping. Cut distance is used to find the number of clusters and clustered objects. The results are verified with Cophenetic correlation coefficient validation index. Hierarchical clustering method suffers from its inability to perform adjustment, once a merge or split decision has been executed. This limitation can be overcome by the integration of hierarica1 agglomeration with iterative relocation methods, as recent studies have emphasized. So, we plan to implement hierarchical clustering combining with other techniques and also to implement the parallel version of it, in future.

6. REFERENCES

[1] Akinobu Sugiyama., Manabu Kotani., 2002, Analysis of gene expression Data Using Self-organizing Maps and k-means Clustering, IEEE, 1342-1345.

[2] Alon. U, Barkai, D.A., Notterman, K., Gish, S., Ybarra, D., Mack, and Levine, A.J., 1999, Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays, In Proc. Natl. Academy of Sciences, 96, 6745-6750.

[3] Ao. S.I, Kevin Y.P, Michael Ng, David Cheung, Fong .P, Ian Melhado and Sham. C., 2005, CLUSTAG: hierarchical clustering and graph methods for selecting SNPs, Oxford University press, 21(5), 1735-1736.

[4] Bandyopadhyay, S., and Maulik, U., 2002. An evolutionary technique based on K-means algorithm for optimal clustering in RN, Information Science, 146, 221–237.

[5] Cheng Y., Church GM., 2000, Biclustering of expression data. Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB), 8:93–103, 2000.

[6] Chen, C.Y., and Ye, F., 2004. Particle swarm optimization algorithm and its application to clustering analysis. In Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control, 789–794.

[7] Clark, F., Olson, 1995, Parallel algorithms for hierarchical clustering Parallel Computing, 21, 1313-1325

[8] Day, W.H,E., and Edelsbrunner, H., 1984, Efficient algorithms for agglomerative hierarchical clustering methods, J. Classification, l(1), 7-24.

[9] Defays, D., 1977, An efficient algorithm for a complete link method, Comput. J, 20, 364-366

[10] Du. Z, Lin. F, 2005, A novel parallelization approach for hierarchical clustering. Parallel Computing, 31, 523-527.

[11] Duran B. S. and Odell. B. S., 1974, Cluster Analysis, A Survey, volume 100 of Lectures Notes in Economics and Mathematical Systems. Springer.

[12] Eisen M., Spellman P., Brown P., Botstein D., 1998, Cluster analysis and display of genome-wide expression patterns. In Proc Natl. Acad. Science USA, 95(25), 14863-14868.

[13] Eisen, M.B., Brown, P.O., 1999, DNA arrays of gene expression, In: methods in enzymology, 303, 179-205.

[14] Getz G., Levine E., and Domany E.,, 2000, Coupled two-way clustering analysis of gene microarray data. In Proc. Natl. Acad. Sci. USA, 97(22), 12079–12084.

[15] Han, J.W., and Kamber. M., 2001, Data Mining Concepts and Techniques. Higher Education Press, Beijing.

[16] Hisashi Koga., Tetsuo Ishibashi., ToshinoriWatanabe., 2007, Fast agglomerative hierarchical clustering algorithm using Locality-Sensitive Hashing, Knowledge Inf. Syst. 12(1), 25–53

[17] Huang, Michael. K. Ng,, 1999, A fuzzy k-modes algorithm for clustering categorical data. IEEE Trans. Fuzzy Systems. 7(4), 446-452.

[18] Jain, A. K., Murty, M. N., and Flynn, P. J., 1999, Data clustering: A review. ACM Computing Surveys, 31(3):264-323.

[19] Karypis G., Han E., Kumar V., 1999. CHAMELEON: hierarchical clustering using dynamic modeling. IEEE Comput. 32(8), 68–75.

[20] Kohonen, T. 1990, The Self-Organizing Map, Proc. IEEE,78(9),1464–1479.

[21] Kennedy, J., and Eberhart, R. C., 1995, Particle swarm optimization. In proceedings of the IEEE International Joint Conference on Neural Network, 4, 1942–1948.

[22] Le. S.Q., Ho.T.B., 2003, A K-sets clustering algorithm for categorical and mixed data, In. Proc of the 6th SANKEN. Int. Symbosium, 124-128.

[23] Luo F, Tang K, Khan L., 2003, Hierarchical clustering of gene expression data. Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering.

[24] Marcilio CP de Souto, Ivan G Costa, Daniel SA de Araujo, Teresa B Ludermir, and Alexander Schliep, 2008, Clustering cancer gene expression data: a comparative study, BMC Bioinformatics, 9, 497.

[25] Minsoo Lee, Yun-mi Kim, Yearn Jeong Kim, Yoon- kyung Lee, and Hyejung Yoon , 2007, An Ant-based Clustering system for Knowledge Discovery in DNA Chip Analysis Data , In Proc of WASET, 23, 261-266.

[26] Murthy, C. A., & Chowdhury, N., 1996 . In search of optimal clusters using genetic algorithms. Pattern Recognition Letters, 17, 825–832.

[27] Paterlini, S., and Krink, T., 2006, Differential evolution and particle swarm optimization in partitional clustering, Computational Statistics and Data Analysis, 50, 1220–1247

[28] Raja Loganantharaj, Satish Cheepala, and John Clifford, 2006, Metric for Measuring the Effectiveness of Clustering of DNA Microarray Expression, BMC Bioinformatics; 7(2), S5.

[29] Sathiyabhama, B., Gopalan, 2006, N.P., Enhanced Correlation Search Technique for Clustering Cancer Gene Expression data, WSEAS Transactions on Information Science and Applications 12(3), 2477- 2484.

[30] Shuanhu Wu1,, Alan Wee Chung Liew2., and Hong Yan3., 2005, OPTOC-Based Clustering Analysis of Gene Expression Profiles in Spectral Space, Springer-Verlag Berlin Heidelberg, LNCS 3498, 709-718.

[31] Sibson, R., 1973, SLINK : An Optimally Efficient Algorithm for the Single Link Cluster Method. Computer Journal, 16, pages 30–34, 1973.

[32] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S and Golub, T.R., 1999, Interpreting Patterns of Gene Expression With Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation, Proc. Natl. Academy of Sciences, 96(6), 2907- 2912.

[33] Vinvent S. Tseng and Ching-Pin Kao, 2005, Efficiently Mining Gene Expression Data via a Novel Parameter less Clustering Method, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2(4), 355–365.

[34] Yi-Tung Kao, Erwie Zahara, I-Wei Kao, 2008, A hybridized approach to data clustering, Expert Systems with Applications, 34, 1754–1762