IJBR
International Journal of Bioinformatics Research
ISSN: 0975–3087, E-ISSN: 0975–9115, Vol. 3, Issue 2, 2011, pp-207-213
Available online at http://www.bioinfo.in/contents.php?id=21

CLUSTER ANALYSIS OF MICROARRAY DATA BASED ON SIMILARITY MEASUREMENT

SOUMEN KR. PATI[1]* AND ASIT KR. DAS[2]
[1]St.Thomas College of Engineering and Technology, 4, D.H. Road, Kolkata, 23
[2]Bengal Engineering and Science University, Shibpur, Howrah, 03, India
*Corresponding Author: Email- soumen_pati@rediffmail.com, akdas@cs.becs.ac.in

Received: September 02, 2011; Accepted: September 16, 2011

Abstract- DNA microarray technology is a fundamental tool in gene expression data analysis. The collection of datasets from the technology has underscored the need for quantitative analytical tools to examine such data. Due to the large number of genes and complex gene regulation networks, clustering is a useful exploratory technique for analyzing these data. Many clustering algorithms have been proposed to analyze microarray gene expression data, but very few of them evaluate the quality of the clusters. In this paper, a novel cluster analysis technique has been proposed without considering number of clusters a priori. The method computes a similarity measurement function based on which the clusters are merged and subsequently splits a cluster by computing the degree of separation of the cluster. The process of splitting and merging performs iteratively until the cluster validity index (i.e. DB index) degrades. The experimental result shows that the proposed cluster analysis technique gives comparable results on gene cancer dataset with existing methods. This study may help raise relevant issues in the extraction of meaningful biological information from microarray expression data.

Keywords- DNA microarray data, Clustering technique, Cluster validity index, Similarity measurement, Splitting and Merging, Cancer data analysis

1. INTRODUCTION

Gene expression microarrays provide a popular technique to monitor the relative expression of thousands of genes under a variety of experimental conditions. In spite of the enormous potential of this technique, challenging problems remain associated with the acquisition and analysis of microarray data [4, 25] that may have a profound influence on the interpretation of the results.

Gene microarray data have provided the opportunity to measure the expression level [2- 4] of thousands of genes simultaneously and this kind of high-throughput data has a wide application in bioinformatics research. In DNA microarray data analysis, for example, biologists measure the expression levels of genes (thousands of them) in the tissue samples from patients, and seek explanations about how the genes of patients relate to the types of cancers they had. Many genes could strongly be correlated to a particular type of cancer. However, biologists prefer to focus on a small subset of genes that dominates the outcomes before conducting in-depth analysis and expensive experiments with a larger set of genes. Therefore, automated discovery of this small subset known as feature selection [5, 6] is highly desirable. A typical microarray data matrix contains the expression levels of thousands of genes across different experimental samples. DNA microarray technology has directed the focus of computational biology [7] towards analytical data interpretation [8]. However, when examining microarray data, the size of the data sets and noise contained within the data sets compromises precise qualitative and quantitative analysis [9]. A standard objective of microarray data analysis is to better understand the gene-to-gene interactions that take place amongst the entire gene pool.

Gene data clustering plays a vital role in microarray (gene) data analysis and computer vision. It is often used to partition a microarray data into separate regions. Gene data clustering [1,10,15] when viewed as a clustering problem aims to partition the given microarray data into clusters such that some rows of microarray data within a cluster are homogeneous whereas the rows from different clusters are heterogeneous with respect to some similarity measure. Some popular clustering algorithms such as k-means [10-13], fuzzy clustering [10, 16, 17], particle swarm optimization (PSO) [18, 19] and mixture model [14] are often used in gene data clustering. All these algorithms try to minimize the within-cluster (average intra-cluster) variance or maximize the inter-cluster separation. These traditional techniques are failed to handle noisy data properly.

In the paper, a novel cluster analysis method has been proposed for partitioning DNA microarray data sets which is accurate and also robust in noisy environment. The method initially generates sufficiently large number of clusters by k-means algorithm [10-13] and then introduces merging and splitting procedure on the clusters. In each iteration, first merging operation occurs successively until a certain condition (explained below) is satisfied and then splitting operation is invoked once. The process terminates while no merging and splitting occurs in two consecutive iterations.

In merging procedure, similarity between every pair of clusters are measured using the proposed criterion function and the cluster pair with maximum similarity are merged together. After merging, average Davies-Bouldin index (DB index) [20-22] of clusters are computed and compared with that obtained before merging. If the new value is less than or greater by a small threshold than the old value, the merging process is continued, otherwise, rollback the process to obtain previous set of clusters.

In splitting procedure, intra cluster distance is computed to measure the dispersion of the objects in the clusters. The clusters with intra cluster distance greater than a threshold are split into three disjoint clusters as follows:

(i) Consider mean and two most distant objects of the cluster as three initial clusters.

(ii) An object is placed to one of the three clusters to which it is nearest.

The proposed clustering methodology does not assume any particular underlying distribution of the data set being considered. Computationally it is simple like the k-means algorithm. On the other hand, it should not sufferer from the limitation of the traditional clustering algorithms which may fail in noisy environment.

The article is organized into four sections. Section 2 gives the detailed description of the proposed cluster analysis methodology. The performance of the proposed method is evaluated in Section 3 using a variety of gene data (microarray data) and compared the results with k-means clustering technique. Finally, conclusions are drawn in Section 4.

2. CLUSTER ANALYSIS

Cluster analysis partitions gene data into meaningful clusters which capture the natural structure of the data to find genes that have similar functionality. For understanding the distribution of data objects in a group, cluster analysis has long been applied in diverse field starting from information science to social sciences and more importantly recently in biological science [23]. Cluster analysis [10, 11, 14] groups gene data based on the available information describing their relationships. The goal of clustering is to group the similar (or related) genes in a cluster and dissimilar (or unrelated) genes in different clusters. Clustering partitions gene microarray data into certain number of groups of similar genes by using a similarity function explores natural structure and identifies interesting patterns in data. Many interesting algorithms [1, 10, 14] are applied to analyze very large datasets but a comprehensive criterion, which would be independent of the final aim of the clustering, has not been formulated yet. Consequently, the clusters become unstable with slight variation of the parameters used in the algorithms. Therefore, the issue closely related to cluster analysis is validation of clusters. However, most algorithms don't provide any means for its validation and evaluation. So it is very difficult to conclude which are the best clusters and should be taken for next step of data processing. The best criterion is greatly dependent on the final aim of the clustering and provided by the users to meet their requirements. There are validity indices [20-22] proposed by the researchers that quantize the goodness of a partition by measuring membership distributions, entropies of the partition, compactness of clusters and others.

In the paper, clusters are analyzed by proposed merging and splitting technique and goodness of the partitions are measured with the help of DB-index (Davies-Bouldin index) [20, 22]. The method initially generates sufficiently large number of clusters by k-means algorithm [10-13] on microarray gene expression data sets and then introduces merging and splitting procedure on the clusters. Here, cluster validity index namely, DB-index is computed and used after merging process to measure the goodness of the clusters which finally helps to obtain optimum set of clusters. The computation of DB-index is for a set of cluster is described briefly below:

It is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. Let $C_1, C_2, \dots\dots, C_k$ be the k number of clusters and then DB-index is defined by equation (1).

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{j=1,2,\dots,k \; and \; i\neq j}\left(\frac{\delta_i^2 + \delta_j^2}{d_{ij}^2}\right) \qquad (1)$$

Where, δ_i^2 and δ_j^2 are the variance of clusters C_i and C_j, respectively and d_{ij}^2 is the distance of cluster centers between C_i and C_j. As a low scatter/variance and high distance between clusters lead to low value of R_{ij}, low value of DB corresponds to clusters that are compact and centers are far away from each other. So the smaller the DB-index better is the clustering [20, 22].

In each iteration, first merging operation occurs successively until a certain condition (explained below) is satisfied and then splitting operation is invoked once. The process terminates while no merging and splitting occurs in two consecutive iterations. The overall proposed method is described in Fig. 1.

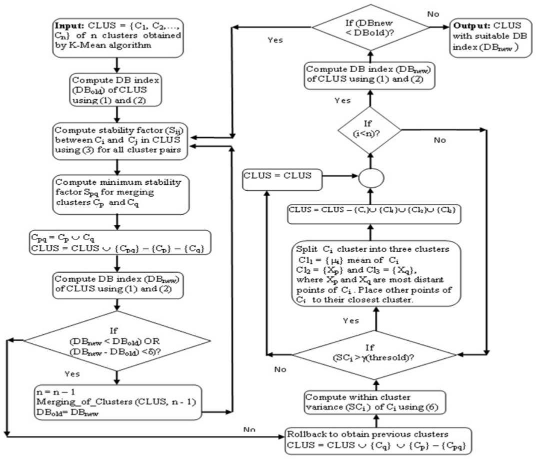

Input: CLUS = {C1, C2,…, Cn} of n clusters obtained by K-Mean algorithm
Compute DB index (DBold) of CLUS using (1) and (2)
Compute stability factor (Sij) between Ci and Cj in CLUS using (3) for all cluster pairs
Compute minimum stability factor Spq for merging clusters Cp and Cq
Cpq = Cp ∪ Cq
CLUS = CLUS ∪ {Cpq} − {Cp} − {Cq}
Compute DB index (DBnew) of CLUS using (1) and (2)
If (DBnew < DBold) OR (DBnew - DBold) <δ)?
Yes
n = n − 1
Merging_of_Clusters (CLUS, n - 1)
DBold= DBnew
No
Rollback to obtain previous clusters
CLUS = CLUS ∪ {Cq} ∪ {Cp} − {Cpq}
Compute within cluster variance (SCi) of Ci using (6)
If (SCi > γ(thresold)?
No
Yes
Split Ci cluster into three clusters
Cl1 = {μi} mean of Ci
Cl2 = {Xp} and Cl3 = {Xq},
where Xp and Xq are most distant points of Ci . Place other points of Ci to their closest cluster.
CLUS = CLUS – {Ci}∪ {Cl1}∪ {Cl2}∪ {Cl3}
CLUS = CLUS
If (i<n)?
No
Yes
Compute DB index (DBnew) of CLUS using (1) and (2)
If (DBnew < DBold)?
Yes
No
Output: CLUS with suitable DB index (DBnew)

Fig. 1- Illustrate flow diagram of proposed method

2.1 Merging of Clusters

Data clustering for a gene data set partitions the data into different groups so that similar nature genes are in a single group. Similarity of genes can be measured based on different features (samples). Combination of them can be used to represent a row (gene) of micro array data [3, 24]. Thus for each row (gene), a feature vector X is associated. Initially, sufficiently large number of clusters is formed on the set of feature vector X by K-means algorithm [10-13] with the assumption that actual number of clusters are very less compare to it. Suppose, there are *n* clusters for a data set and its DB index is DB_n, obtained using (1). Then, for any two clusters C_i and C_j (*i, j* =1, 2,…..,*n* and *i ≠ j*) similarity factor s_{ij} is computed using (2) with the implication that, smaller the s_{ij} closer the clusters are and vice versa.

$$s_{ij} = \frac{|\delta_i^2+\delta_j^2-\delta_{ij}^2|\times d_{ij}^2}{N} \qquad (2)$$

Where, *N* is a constant used as normalization factor, δ_i^2 and δ_j^2 are the variance of clusters C_i and C_j respectively, δ_{ij}^2 is the variance of cluster C_{ij} obtained combining clusters C_i and C_j computed using (3) and $d_{ij}^2 = \left||\mu_i - \mu_j|\right|^2$ gives the distance between the cluster centers μ_i and μ_j.

$$\delta_{ij}^2 = \frac{n_i\times(\delta_i^2+d_i^2)+n_j\times(\delta_j^2+d_j^2)}{n_i+n_j} \qquad (3)$$

Where, n_i and n_j denote the number of objects in clusters C_i and C_j respectively, $d_i^2 = \left||\mu_i - \mu_{ij}|\right|^2$ gives the distance between clusters C_i and C_{ij} and $d_j^2 = \left||\mu_j - \mu_{ij}|\right|^2$ gives the distance between clusters C_j and C_{ij}. The weighted mean μ_{ij} representing the center of combined cluster C_{ij} is computed using (4).

$$\mu_{ij} = \frac{(n_i\times\mu_i)+(n_j\times\mu_j)}{n_i+n_j} \qquad (4)$$

From (2) it is noticed that, if the difference between the sum of the variance of two clusters and the variance of the cluster obtained by the combination of two clusters is low and at the same time the distance between two individual cluster center is low, then the similarity factor between the cluster is low and the clusters are closed to each other. This implies that less the similarity factor between the cluster pair more compact the clusters are.

Thus, a similarity matrix S = $(s_{ij})_{n\times n}$ is generated using (2) which is a symmetric matrix with empty diagonal entries, as the similarity of a cluster with itself is not required. So, $\frac{n(n-1)}{2}$ similarity factors among each possible cluster pairs stored above the leading diagonal of S carry the information based on which clusters are merged.

In each iteration, only the cluster pairs with lowest similarity factor are merged reducing number of clusters by one. As a result, (n – 1) clusters are obtained whose DB index DB_{n-1} is computed using (1). The process terminates if DB_{n-1} is large enough then DB_n and the system is roll backed to the previous state to preserve the previous set of *n* clusters; otherwise same process is repeated with (n – 1) clusters. The detail algorithm for merging process is given below:

Algorithm: Merging_of_Clusters (CLUS, n)

Input: CLUS = {C_1, C_2, …, C_n} of n clusters obtained by K-Mean algorithm
Output: Set of clusters in CLUS after merging

```
Begin
 $DB_{old}$ = DB index of CLUS using (1)
 For i = 1 to n {
    For j = i+1 to n {
       $S_{ij}$ = Similarity factor between $C_i$ and $C_j$ in CLUS
             using (2)
    }
 }
  /*compute minimum stability factor and corresponding
clusters*/
 min = $S_{12}$
 For i = 1 to n {
     For j = i+1 to n {
        If ($S_{ij}$ < min) {
           min = $S_{ij}$
           p = i
           q = j
        }
     }
 }
 $C_{pq} = C_p \cup C_q$
 CLUS = CLUS $\cup$ {$C_{pq}$} – {$C_p$} – {$C_q$}
 $DB_{new}$ = DB index of CLUS using (1)
  If (($DB_{new} < DB_{old}$) | | (($DB_{new} - DB_{old}$) < $\delta$)))  {
     /* $\delta$ > 0, a small threshold value*/
     n = n – 1
     Merging_of_Clusters (CLUS, n - 1)
  }
  Else {    /*rollback to obtain previous clusters*/
     CLUS = CLUS $\cup$ {$C_q$}  $\cup$ {$C_p$} – {$C_{pq}$}
     Return CLUS
 }
End
```

2.2 Splitting of a cluster

The data set is initially clustered by K-Mean's algorithm with a large value of K. After it the clusters are only tried to merge using merging algorithm described in section 2.1. So there is a high possibility that the objects are situated in scatter manner within the clusters. Such clusters are known as sparse clusters which need to be split into various clusters. In the paper, a sparse cluster is split into three clusters considering centroid of the sparse cluster and two most distant objects as their centers. The other objects of the sparse cluster are placed in one of the three clusters to which they are closest. The variance *SC_i* of the objects in a cluster C_i is measured by their variance, computed using (5).

$$SC(i) = \sum_{\forall x \in C_i} \frac{d(x-\mu)}{|C_i|} \qquad (5)$$

Where, μ is the center of the cluster and $d(x - \mu)$ is the Euclidian distance between an object x and center of the cluster. Each of the clusters with variance greater than a threshold (γ) is split into three clusters. The splitting algorithm is described below:

Algorithm: Splitting_of_Clusters (CLUS, n)

Input: CLUS = {C_1, C_2,… ,C_n} of n clusters obtained after merging
Output: Set of clusters in CLUS after splitting

```
Begin
  For i = 1 to n {
        /* mean or center computation*/
      Sum = 0;
      For j = 1 to |$C_i$| {
           Sum = Sum + $X_j$
      }
  $\mu_i$ = Sum / |$C_i$|   /* $\mu_i$ is the mean or center of cluster $C_i$ */
      /* scatter or variance within cluster */
    Sum = 0;
    For j = 1 to |$C_i$| {
       Sum = Sum + ||$X_j$ - $\mu_i$||
    }
   $SC_i$ = Sum / |$C_i$|
  /* if computed scatter is greater than threshold ($\gamma$), cluster $C_i$ splits into three clusters $Cl_1$, $Cl_2$ and $Cl_3$ */
  If ($SC_i$ >$\gamma$) {
    ||$X_p$ - $X_q$ || = $\max_{1\le i\le|C_i|} \max_{1\le j\le|C_i|, i\ne j} || X_i - X_j ||$
    $Cl_1$ = {$\mu_i$}
    $Cl_2$ = {$X_p$}
    $Cl_3$ = {$X_q$}
    For j = 1 to |$C_i$| {
       If (||$X_j$ - $\mu_i$||==min (||$X_j$ - $\mu_i$||, ||$X_j$ - $X_p$||, ||$X_j$ -   $X_q$||)
                 $Cl_1$ = $Cl_1$ $\cup$ {$X_j$}
       Else if (||$X_j$ - $X_p$||==min (||$X_j$ - $\mu_i$||,||$X_j$ - $X_p$||,||$X_j$ - $X_q$||))
                 $Cl_2$ = $Cl_2$ $\cup$ {$X_j$}
              Else $Cl_3$ = $Cl_3$ $\cup$ {$X_j$}
         }
         CLUS = CLUS – {$C_i$}$\cup$ {$Cl_1$}$\cup$ {$Cl_2$}$\cup$ {$Cl_3$}
     }
  }
End.
```

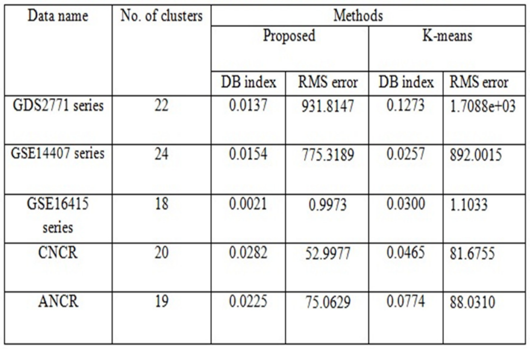
3. EXPERIMENTAL EVALUATION

Experimental studies presented here provide an evidence of effectiveness of proposed microarray data cluster analysis. The five microarray gene database used in the experiment are described below where each row in the data sets represents the expression pattern of one gene and each column represents an experimental sample. Experiments were carried out on large number of different kinds of microarray data (cancerous data), few of them [24, 25] described below are summarized. Each dataset contains two types of samples, one group is normal and other is cancerous.

a. **GDS2771 series data:** There exist 2000 rows (genes) and 72 columns (samples). Among these, 36 samples are normal and others are cancerous.
b. **GSE14407 series data:** It contains 54675 rows (genes) and 24 samples (12 ovarian surface epithelial cells and 12 laser capture micro deselected serous papillary ovarian cancers).
c. **GSE16415 series data:** It contains 32878 rows (genes) and 10 samples (5 diabetic and 5 control women samples).
d. **Carcinoma Normal dataset Cancer Research data (CNCR):** This is a well-understood gene data base. It contains 7457 rows (genes) and 36 samples (18 samples are normal and others are cancerous).
e. **Adenomas Normal Cancer Research data (ANCR):** This is a well-understood gene data base. It contains 7086 rows (genes) and 8 samples (4 samples are normal and others are cancerous).

The proposed technique initially generates large number of clusters by applying k-means algorithm with *k* over the range [40 – 60] and observes that the final results are not significantly changed (in terms of validity index measure). The clustering results described here are achieved by considering k = 50 initial number of clusters. After successive iteration of merging and subsequent splitting process discussed in section 2, final number of cluster is obtained, listed in Table 1. Considering same number of clusters as the value of k, k-means algorithm is applied on the data sets and a comparison is made in Table 1 between the proposed and k-means algorithm by evaluating DB-index and RMS error in cluster sets. All the algorithms are implemented using Mat lab 7.8.1 version. The comparison is performed on PC (Intel(R) Core(TM) 2 Duo T5750 2.0 GHz, 2.0 GHz with 2.0 GB of Ram).

Table 1: Comparison of clusters based on proposed and k-means algorithm



Data name	No. of clusters	Methods			
		Proposed		K-means	
		DB index	RMS error	DB index	RMS error
GDS2771 series	22	0.0137	931.8147	0.1273	1.7088e+03
GSE14407 series	24	0.0154	775.3189	0.0257	892.0015
GSE16415 series	18	0.0021	0.9973	0.0300	1.1033
CNCR	20	0.0282	52.9977	0.0465	81.6755
ANCR	19	0.0225	75.0629	0.0774	88.0310

The results show that DB index and RMS errors produced by the proposed method are less than that produced by the K-mean clustering methods for different five cancerous microarray data sets, which confirms the potentiality and superiority of the proposed method. The method changes number of clusters and corresponding DB-index in each iteration, the nature of which can be visualized by Fig. 2 to Fig. 6 for five mentioned data set.

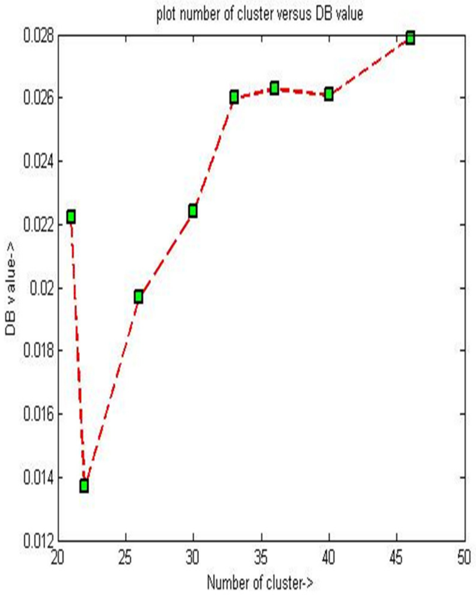


Fig. 2-GDS2771 series data set

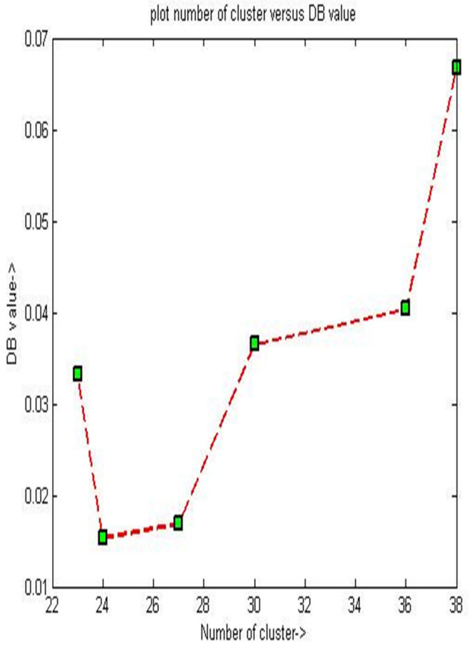


Fig. 3- GSE14407 series data set

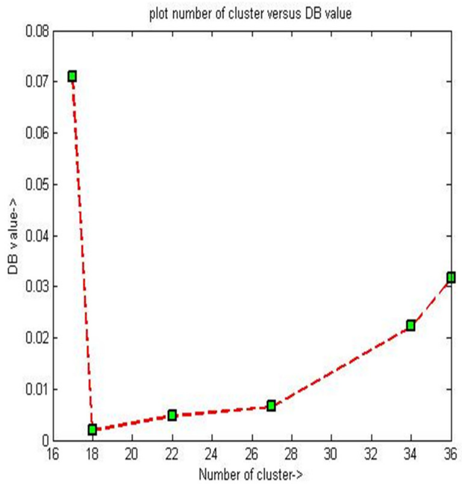


Fig.4- GSE16415 series data set

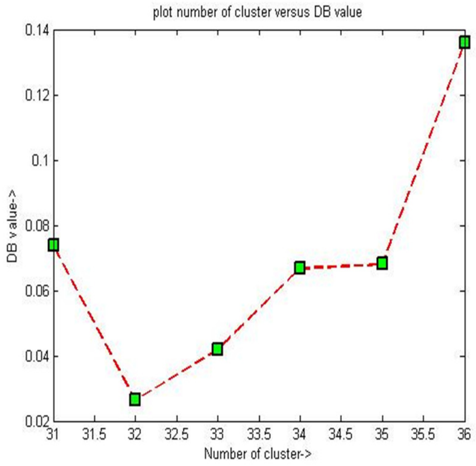


Fig. 5- CNCR data set

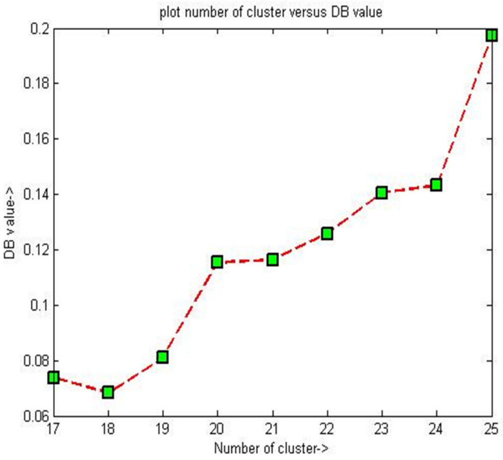


Fig. 6- ANCR data set

From the figures, it is observed that, DB value decreases as number of clusters reduces and for further reduction DB value increases, which terminates the process and gives optimal set of clusters. For example, in Fig. 2, number of clusters is gradually decreases from 50 to 18 with decreasing DB value, and then DB value increases, which gives total 18 clusters for GDS2771 series data set as optimal set of clusters. The process always regenerates the clusters and at the same time number of clusters is reduced. For example, number of clusters reduces to 46 from 50 after first iteration for GDS2771 series data set, shown in Fig. 2. Now k-means algorithm is applied with k = 46 and DB-index is computed for both the cluster sets obtained by the proposed and k-mean algorithms. This comparison is made in each iteration until the optimal set of clusters is found, shown in Fig. 7. Similar comparison is made for other data set from Fig. 8 to Fig. 11.

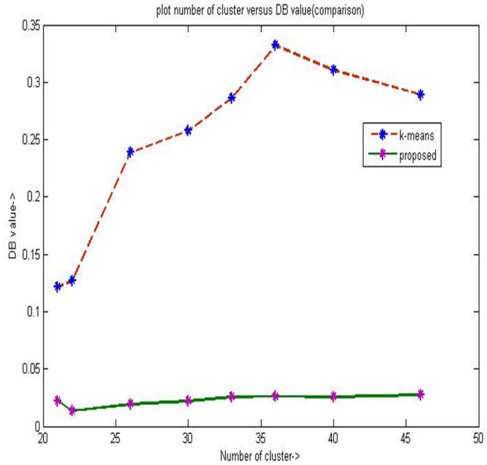


Fig. 7- DB-index comparison for GDS2771

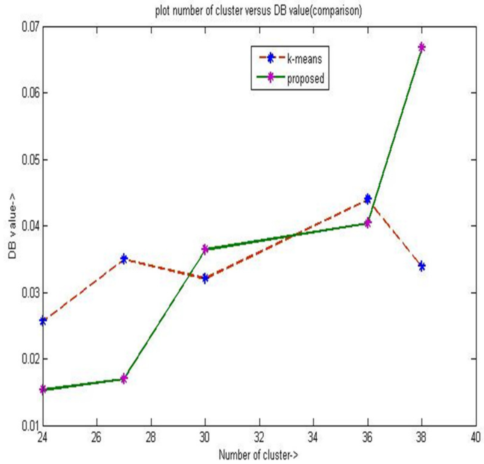


Fig. 8- DB-index comparison for GSE14407

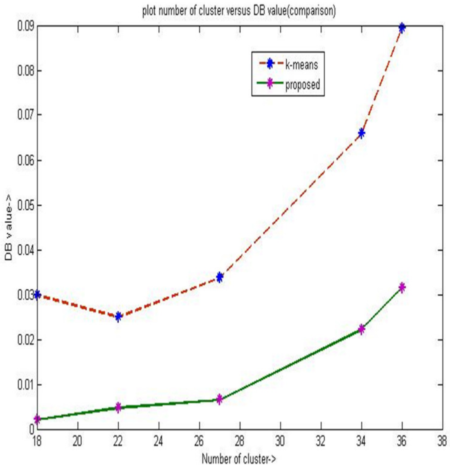


Fig. 9- DB-index comparison for GSE16415

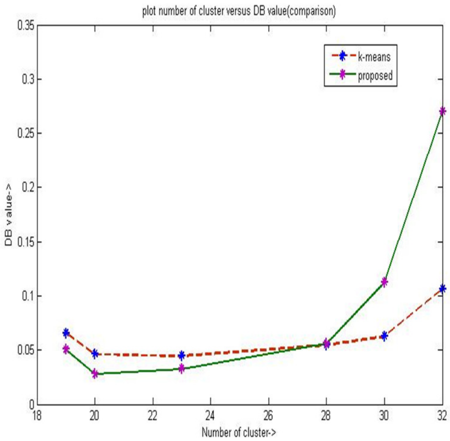


Fig. 10- DB-index comparison for CNCR

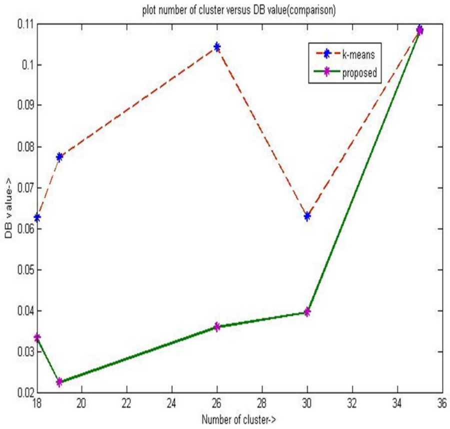


Fig. 11- DB-index comparison for ANCR

4. DISCUSION AND CONCLUSIONS

In the paper a novel cluster validation technique have been proposed for obtaining optimal set of clusters based on a new cluster similarity measure. Initially large numbers of clusters are generated using k-means algorithm, and then these clusters are successively merged and split based on their degree of compactness and separation. Experimental results shown for five different kinds of microarray cancerous data evaluates the performance of the proposed algorithm both qualitatively as well as quantitatively. Comparative study is made with traditional unsupervised algorithms namely k-means with respect to DB index and Root Mean Square error (RMS) which shows that the proposed method performs fairly well in terms of the clustering quality.

References

[1] Theodoridis S. and Koutroumbas K. (2003) *Pattern Recognition Amsterdam: Elsevier Academic Press*.

[2] Zhang M.Q. (1999) *Genome Res*, 9:681-8.

[3] Arhondakis S., Auletta F., Torelli G., D'Onofrio G. (2004) *Base composition and expression level of human genes, Gene*, 325, 165-169.

[4] Alon A., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D., and Levine A.J. (1999) *Proc. Natl. Acad. Sci.*, 1, 6745–6750.

[5] Ding C. and Peng H.C. (2003) *Proc. Second IEEE Computational Systems Bioinformatics Conf.*, 523–528.

[6] Yu L. and Liu. H. (2004) *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining,* 737 – 742.

[7] Slonim D. K., Tamayo P., Mesirov J. P., Golub T.R. and Lander E.S. (2000) *Proceedings of the Forth Annual Conference on Research in Computational Molecular Biology*, 263–272.

[8] Muralidhar K. and Sarathy R. (1999) *ACM Trans. Database Syst.*, 24(4), 487–493.

[9] Petrov A. and Shams S. (2004) *Microarray image processing and quality control, VLSI Signal Processing*, 38(3), 211–226.

[10] Qu Y. and Xu S. (2004) *Supervised cluster analysis for microarray data based on multivariate Gaussian mixture, Bioinformatics,* 20, 1905-13.

[11] Guha S., Rastogi R. and Shim K. (1998) *Proc. of ACM SIGMOD International Conference on Management of Data*, 73 – 84.

[12] Bradley P.S., Bennett K.P. and Demiriz A. (2000) *Constrained k-means clustering (Technical ReportMSR-TR-2000-65), Microsoft Research, Redmond, WA*.

[13] Suresh R. M., Dinakaran K. and Valarmathie P. (2009) *Model based modified k-means clustering for microarray data*, *d.o.i.* 10.1109/ICIME.2009.53, 271-273.

[14] Xu R. and Wunsch D. (2005) *IEEE Trans. Neural Networks*, 16(3), 645-678.

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

[16] Bezdek J.C. (1981) *Pattern recognition with fuzzy objectivc function algorithms. New York: Plenum Press*.

[17] Bertoni A. and Giorgio V. (2007) *Fuzzy ensemble clustering for DNA microarray data analysis, Lecture Notes in Computer Science*, 3931, 537-543.

[18] Kennedy J. and Eberhart R.C. (1995) *Proceedins of the IEEE International Joint Conference on Neural Networks*, 4, 1942-1948.

[19] Kennedy J., Eberhart R.C. and Shi Y. (2002) *Swarm Intelligence, Morgan Kaufmann*.

[20] Davies D.L. and Bouldin D.W. (1989) *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2), 95 – 104.

[21] Halkidi M., Batistakis Y. and Vazirgiannis M. (2002) *Cluster validity methods: part II, SIGMOD Rec.*, 31(3),19 –27.

[22] Halkidi M. and Vazirgiannis M. (2001) *Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set, Proc. of ICDM*, 187 – 194.

[23] P´erez-Jim´enez M.J. and Romero-Campero F.J. (2006) *P systems, a new computational modeling tool for systems biology*, 4220, 176–197.

[24] Stanford Microarray Database: http://smd.stanford.edu.

[25] Cancer Definition: http://www.nlm.nih.gov/medlineplus/medlineplus.html.