IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 3, May 2010
ISSN (Online): 1694-0784
ISSN (Print): 1694-0814

An Effective Technique for Clustering Incremental Gene Expression data

Sauravjyoti Sarmah[1] and Dhruba K. Bhattacharyya[2]

[1,2]Department of Computer Science & Engg., Tezpur University[3]
Tezpur-784028, Assam, India

Abstract

This paper presents a clustering technique (GenClus) for gene expression data which can also handle incremental data. It is designed based on density based approach. It retains the regulation information which is also the main advantage of the clustering. It uses no proximity measures and is therefore free of the restrictions offered by them. GenClus is capable of handling datasets which are updated incrementally. Experimental results show the efficiency of GenClus in detecting quality clusters over gene expression data. Our approach improves the cluster quality by identifying sub-clusters within big clusters. It was compared with some well-known clustering algorithms and found to perform well in terms of the z-score cluster validity measure.

Keywords: Clustering, microarray, gene expression, density based, incremental.

1. Introduction

According to [1], most data mining algorithms developed for microarray gene expression data deal with the problem of clustering. Clustering genes groups similar genes into the same cluster based on a proximity measure. Genes in the same cluster have similar expression patterns. One of the characteristics of gene expression data is that it is meaningful to cluster both genes and samples. Co-expressed genes can be grouped into clusters based on their expression patterns [2] and [3]. In *gene-based* clustering, the genes are treated as the objects, while the samples are the features. In *sample based* clustering, the samples can be partitioned into homogeneous groups where the genes are regarded as features and the samples as objects. Both the gene-based and sample based clustering approaches search exclusive and exhaustive partitions of objects that share the same feature space. The third category, that is *subspace clustering*, captures clusters formed by a subset of genes across a subset of samples. For subspace clustering algorithms, either genes or samples can be regarded as objects or features. The details of the challenges and the representative clustering techniques will be discussed in *Section* 2. The result of clustering is dependant on the proximity measure used [4] and different measures give different results. In this paper, we introduce an effective gene-based clustering approach (GenClus), which is capable of identifying clusters and sub-clusters of arbitrary shapes of any gene expression dataset, even in presence of noise. GenClus attempts to find sub-clusters which may be relevant for biologists. It does not use any proximity measure during clustering the genes and therefore free from the restriction offered by various proximity measures. GenClus gives a hierarchical view of the clusters and sub-clusters formed. With the increasing development of internet technology and with the constant increase in the microarray experimentation conducted, it has led to the ever-increasing volume of data. There is therefore a need to introduce incremental clustering so that updates can be clustered in an incremental manner. To handle such increase in volume of microarray data, incremental clustering technique often has been found suitable. This paper also introduces an incremental version of GenClus i.e., InGenClus which has been established to perform well in terms of several gene datasets.

Both GenClus and InGenClus can be found to be significant in view of the following points:

- provides a hierarchical cluster solution;
- free from the use of proximity measures;
- faster processing due to simplified matching mechanism;
- capable of handling noisy datasets;
- does not require the number of clusters apriori;

GenClus improves the quality of the clusters by identifying sub-clusters within large clusters. It can also handle the situation when the database is updated incrementally using less computation time.

[3]The department is funded by UGC's DRS- Phase I under the SAP

In Section 2, we present some gene-based clustering techniques, Section 3 presents our proposed clustering algorithm GenClus and Section 4 presents our proposed incremental version of GenClus (InGenClus). Section 5 reports the performance evaluation of the algorithms and finally Section 6 presents the conclusion. Next, we discuss some of the clustering techniques.

2. Clustering Techniques

The goal of gene-based clustering is to group co-expressed genes together. Co-expressed genes indicate co-function and co-regulation [4]. Gene expression data has certain special characteristics and is a challenging research problem. Here, we will first present the challenges of gene-based clustering and then review a series of gene-based clustering algorithms.

2.1 Challenges of Gene-based Clustering

The purpose of clustering gene expression data is to reveal the natural structure inherent in the data. A good clustering algorithm should depend as little as possible on prior knowledge, for example requiring the pre-determined number of clusters as an input parameter. Clustering algorithms for gene expression data should be capable of extracting useful information from noisy data. Gene expression data are often highly connected and may have intersecting and embedded patterns [5]. Therefore, algorithms for gene-based clustering should be able to handle this situation effectively. Finally, biologists are not only interested in the clusters of genes, but also in the relationships (i.e., closeness) among the clusters and their sub-clusters, and the relationship among the genes within a cluster (e.g., which gene can be considered as the representative of the cluster and which genes are at the boundary area of the cluster). A clustering algorithm, which also provides some graphical representation of the cluster structure is much favored by the biologists.

2.2 Gene based Clustering Techniques: A brief review

A large number of clustering techniques have been reported for analyzing gene expression data. They have been broadly classified into the following approaches.

Partitional Approaches: K-means [6] is a typical partition-based clustering algorithm used for clustering gene expression data. It divides the data into pre-defined number of clusters in order to optimize a predefined criterion. The major advantages of it are its simplicity and speed, which allows it to run on large datasets. However, it may not yield the same result with each run of the algorithm. Often, it can be found incapable of handling outliers and is not suitable to detect clusters of arbitrary shapes. A Self Organizing Map (SOM) [7] is more robust than K-means for clustering noisy data. It requires the number of clusters and the grid layout of the neuron map as user input. Specifying the number of clusters in advance is difficult in case of gene expression data. Moreover, partitioning approaches are restricted to data of lower dimensionality, with inherent well-separated clusters of high density. But, gene expression data sets may be high dimensional and often contain intersecting and embedded clusters. A hierarchical structure can also be built based on SOM such as Self-Organizing Tree Algorithm (SOTA) [8]. Another example of SOM extension is the Fuzzy Adaptive Resonance Theory (Fuzzy ART) [9] which provide some approaches to measure the coherence of a neuron (e.g., vigilance criterion). The output map is adjusted by splitting the existing neurons or adding new neurons into the map, until the coherence of each neuron in the map satisfies a user specified threshold.

Hierarchical Approaches: Hierarchical clustering generates a hierarchy of nested clusters. These algorithms are divided into agglomerative and divisive approaches. Unweighted Pair Group Method with Arithmetic Mean (UPGMA), presented in [3], adopts an agglomerative method to graphically represent the clustered dataset. However, it is not robust in the presence of noise. In [10], the genes are split through a divisive approach, called the Deterministic-Annealing Algorithm (DAA). The Divisive Correlation Clustering Algorithm (DCCA) [11] uses Pearson's Correlation as the similarity measure. All genes in a cluster have highest average correlation with genes in that cluster. Hierarchical clustering not only groups together genes with similar expression patterns but also provides a natural way to graphically represent the data set allowing a thorough inspection. However, a small change in the data set may greatly change the hierarchical dendrogram structure. Another drawback is its high computational complexity.

Density Based Approaches: Density based clustering identifies dense areas in the object space. Clusters are highly dense areas separated by sparsely dense areas. DBSCAN [12] was one of the pioneering density based algorithms used over spatial datasets. In [5], Jiang *et. al.* propose the Density-Based Hierarchical clustering method (DHC) to identify co-expressed gene groups. It can identify embedded clusters in the dataset and can also handle outliers. It can effectively visualize the internal structure of the data set. A kernel density clustering method for gene expression profile analysis is reported in [13]. An alternative to this is to define the similarity of points in terms of their shared nearest neighbors. This idea

was first introduced by Jarvis and Patrick [14]. In [15], a density based gene clustering algorithm, DGC, is presented. DGC uses the regulation information along with the order preserving [16] nature of gene expression data to identify clusters over gene expression data. A density-based approach discovers clusters of arbitrary shapes even in the presence of noise. However, density-based clustering techniques suffer from high computational complexity with increase in dimensionality (even if spatial index structure is used) and input parameter dependency.

Model Based Approaches: Model based approaches provide a statistical framework to model the cluster structure in gene expression data. The Expectation Maximization (EM) algorithm [17] discovers good values for its parameters iteratively. It can handle various shapes of data, but can be very expensive since a large number of iterations may be required. In [18], a signal shape similarity method used to cluster genes using a Variational Bayes Expectation Maximization algorithm [19]. A model-based approach provides an estimated probability that a data object will belong to a particular cluster. Thus, a gene can have high correlation with two totally different clusters. However, the model-based approach assumes that the data set fits a specific distribution which is not always true.

Graph Theoretical Approaches: In graph-based clustering algorithms, graphs are built as combinations of objects, features or both, as nodes and edges, and partitioned by using graph theoretic algorithms. Graph theoretic algorithms are also used for the problem of clustering cDNAs based on their oligo-nucleotide fingerprints [20]. CLuster Identification via Connectivity Kernels (CLICK) [21] is suitable for subspace and high dimensional data clustering. The Cluster Affinity Search Technique (CAST) by [2] takes as input pair-wise similarities between genes and an affinity threshold. It does not require a user-defined number of clusters and handles outliers efficiently. But, it faces difficulty in determining a good threshold value. To overcome this problem, E-CAST [22] calculates the threshold value dynamically based on the similarity values of the objects that are yet to be clustered. In [23], a graph based algorithm for identifying disjoint clusters over gene expression datasets is presented. It is based on the concept that inter-cluster genes have more repulsion between them while the repulsion of intra-cluster genes is less. The cluster results are dependent on a connectivity threshold which is calculated dynamically during the cluster creation process.

Soft Computing Approaches: Fuzzy c-means [24] and Genetic Algorithms (GA) (such as [25] and [26]) have been used effectively in clustering gene expression data. The Fuzzy c-means algorithm requires the number of clusters as an input parameter. The GA based algorithms have been found to detect biologically relevant clusters but are dependent on proper tuning of the input parameters.

The current information explosion, fuelled by the availability of the World Wide Web and the huge amount of microarray experiments being conducted, has led to ever increasing volume of data. Therefore, there is a need to introduce incremental clustering so that updates can be clustered in an incremental manner. Though a lot of research has been performed on incremental clustering in other application domains, incremental clustering of gene expression data has not been exploited much yet.

Incremental Algorithms: In [27], the authors present an incremental clustering approach based on the DBSCAN [12] algorithm. A one pass clustering algorithm for relational datasets is proposed in [28]. Rough set theory is employed in the incremental approach for clustering interval datasets in [29]. In [30], an incremental genetic K-means algorithm is presented. In [31], an incremental gene selection algorithm using a wrapper-based method that reduces the search space complexity since it works on the ranking directly, is presented. In [32], an incremental clustering over gene expression data is presented that uses the regulation information to store the cluster information for use when clustering genes incrementally.

2.3. Discussion

From the discussion above, we conclude that various clustering algorithms require different types of input parameters and clustering results are highly dependent on the values of parameters. Gene expression data has coherent patterns embedded in the full gene space, identification of which is an important research field. Coherent genes may indicate co-regulation and hence fall under the same functional classification. Clustering algorithms that do not require the number of clusters as an input parameter and are robust to noise are of utmost importance. Clustering algorithms are sensitive to the proximity measure chosen. In this paper, we present a gene based clustering technique which is able to identify clusters automatically from the dataset. An incremental version of GenClus is also presented that is capable of handling incremental gene expression datasets.

3. GENCLUS

GenClus is a gene based clustering technique which adopts the notion of density based approach as can be found in [12], [15]. It exploits a discretization technique which retains the up- or down- regulation information. GenClus normalizes the gene expression data and works over a discrete domain (of regulation information). Clustering is then run on the discretized data. The gene expression data is normalized to have mean 0 and standard deviation 1. Expression data having a low variance across conditions as well as data having more than 3-fold variation are filtered. Discretization is then performed on this normalized expression data. Discretization uses the regulation information, i.e. up- or down- regulation in each of the conditions for a particular gene. Here, let G be the set of all genes and T be the set of all conditions. The discretization is done as follows:

i. The discretized value of gene g_i at condition, t_1 (i.e., the first condition)

$$\xi_{g_i,t_1} = \begin{cases} 1 \ if \ \varepsilon_{g_i,t_1} > 0 \\ 0 \ if \ \varepsilon_{g_i,t_1} = 0 \\ -1 \ if \ \varepsilon_{g_i,t_1} < 0 \end{cases}$$

ii. The discretized values of gene g_i at conditions t_j (j = 1,..($T-1$)) i.e., at the rest of the conditions ($T - \{t_1\}$)

$$\xi_{g_i,t_{j+1}} = \begin{cases} 1 \ if \ \varepsilon_{g_i,t_j} < \varepsilon_{g_i,t_{j+1}} \\ 0 \ if \ \varepsilon_{g_i,t_j} = \varepsilon_{g_i,t_{j+1}} \\ -1 \ if \ \varepsilon_{g_i,t_j} > \varepsilon_{g_i,t_{j+1}} \end{cases}$$

where ξ_{g_i,t_j} is the discretized value of gene g_i at conditions t_j (j = 1,..($T-1$)). The expression value of gene g_i at condition t_j is given by ε_{g_i,t_j}. We see in the above computation that the first condition, t_1, is treated as a special case and its discretized value is directly based on ε_{g_i,t_1} i.e., the expression value at condition t_1. For the rest of the conditions the discretized value is calculated by comparing its expression value with that of the previous value. This helps in finding whether the gene is up- (1) or -down (-1) regulated at that particular condition. Each gene will now have a regulation pattern (ρ) of 0, 1, and -1 across the conditions or time points. This pattern is represented as a string.

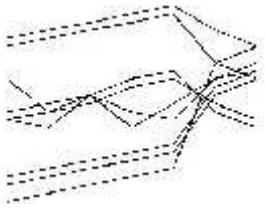

Fig. 1. Expression profiles of an example dataset

Gene 1	Regulation	-1 1 1 1 1 -1 -1
	Range	0 0 0 1 1 0 0
Gene 2	Regulation	-1 1 1 1 1 -1 -1
	Range	0 0 0 1 1 0 0
Gene 3	Regulation	-1 1 1 1 1 1 1
	Range	0 0 0 0 1 2 3
Gene 4	Regulation	-1 1 1 1 1 1 1
	Range	0 0 0 0 1 2 3
Gene 5	Regulation	-1 1 1 1 1 1 1
	Range	0 0 0 0 1 4 4
Gene 6	Regulation	-1 1 -1 -1 1 -1 -1
	Range	0 0 0 0 0 1 1
Gene 7	Regulation	1 1 1 1 1 -1-1
	Range	0 0 0 0 1 0 0
Gene 8	Regulation	1 1 1 1 1 -1 -1
	Range	0 0 0 1 1 -1 -1
Gene 9	Regulation	1 -1 1 -1 1 1 1
	Range	0 -1 0 -1 0 0 0

Fig. 2. Regulation and range information of the example dataset of 1

Each gene is divided into various *range−ids* depending on their expression values as follows. The range value for each expression level is given by uniformly dividing the difference between the maximum and minimum values in the normalized data.

$$range\_value = \frac{Max_{EV} - Min_{EV}}{I}$$

where Max_{EV} is the maximum expression value and Min_{EV} is the minimum expression value. For example, suppose interval, I = 7. Therefore, we will have 7 *range_ids* (3, 2, 1, 0, -1, -2, -3), where the expression values of a gene falling in the corresponding range will get its *range_id*. Now, each gene will have a pattern of *range_id*s across the conditions or time points which is represented as a string. Fig. 2 illustrates an example of a discretized matrix showing the regulation pattern and *range_id*s, where the number of intervals is set to 7, namely (3,2,1,0,-1,-2,-3). The regulation information and range values are used together to cluster the gene expression dataset using a density based approach. A string matching approach is used for matching the regulation pattern and range pattern of two genes. Next, we give some definitions which provide the foundation of GenClus.

Definition 1. *Neighborhood level of a gene*: A gene g_j is said to be a neighbor of gene g_i i.e., $g_j \in N_{level}(g_i)$ if (i) g_i matches with g_j over each of the v conditions, where v is greater than a user defined threshold, α; (ii) $range\_id(g_i, t_k) \pm level = range\_id(g_j, t_k)$, t_k refers to the conditions where $k = 1, 2, \cdots, T$ and *level* is a dynamically calculated parameter. (Initially *level* = 0).

Definition 2. *Core gene*: A gene g_i is said to be a core gene if $| N_{level}(g_i) | \geq \sigma$ (user-defined threshold).
In our experiments, we have obtained good results for $\sigma = 2$. Initially, *level* = 0 and the neighborhood of gene g_i is searched for genes satisfying the core gene condition of *Definition* 2. If no neighbor gene is found, then level is increased in both positive and negative range by one *i.e.*, we search for neighbor genes in adjacent *range_ids* of (g_i, t_k) and the neighborhood search continues.

Definition 3. *Direct-Reachability*: A gene g_j is said to be directly reachable from another gene g_i if g_i is a core gene and $g_j \in N_{level}(g_i)$.

Definition 4. *Reachability*: A gene g_j is said to be reachable from another gene g_i if there is a chain of genes $g_1, g_2, \cdots, g_p$ between g_i and g_j such that g_{i+1} is directly reachable from g_i.

Finding sub-clusters within bigger clusters gives the finer clustering of a dataset. Sub-cluster information may be useful for the biologists by means of visual display and in the interpretation.

Definition 5. *Sub-Cluster*: Let D_G be a database of genes.
A sub-cluster S_i is a non-empty subset of D_G satisfying the following conditions:
1. $\forall g_i, g_j$: if $g_i \in S_i$ and g_j is reachable from g_i, then $g_j \in S_i$.
2. g_i matches with g_j over each of the v conditions.
3. $range\_id(g_i, t_k) \pm level = range\_id(g_j, t_k)$, t_k refers to the conditions where $k = 1, 2, \cdots, T$.

Definition 6. *Cluster*: Gene g_i, $g_j \in C_i$ (i^{th} cluster), if g_i matches with g_j over each of the v conditions i.e., all genes having same regulation pattern over v conditions are grouped into the same cluster.

Subclusters S_j where, $j = 1, 2, \cdots$ will belong to cluster C_i if they have the same regulation pattern.

Definition 7. *Noise Genes*: Let $C_1, C_2, \cdots C_n$ be the set of clusters of D_G, then noise is the set of genes in D_G not belonging to any cluster C_i, i.e.,

$$noise = \{g_x \in D_G \mid \forall i : g_x \notin C_i\}$$

The clustering process starts with an arbitrary gene g_i and searches the neighborhood of it to check if it is core. If g_i is not core then the process restarts with another unclassified gene. If g_i is a core gene, then clustering proceeds with finding all reachable genes from g_i. All reachable genes are assigned the same sub_cluster_id as g_i. From the neighbors of g_i, if any gene satisfies the core gene condition, sub cluster expansion proceeds with that gene. The process continues till no more genes can be assigned to the sub cluster. The process then restarts with another unclassified gene and starts forming the next sub cluster. The clustering process continues till no more genes can be assigned sub_cluster_id. Once all sub clusters have been assigned, the process groups all sub-clusters as well as genes having no sub_cluster_id but having the same regulation pattern into the same cluster and assign them the same cluster_id. All unclassified genes are now termed as noise genes.

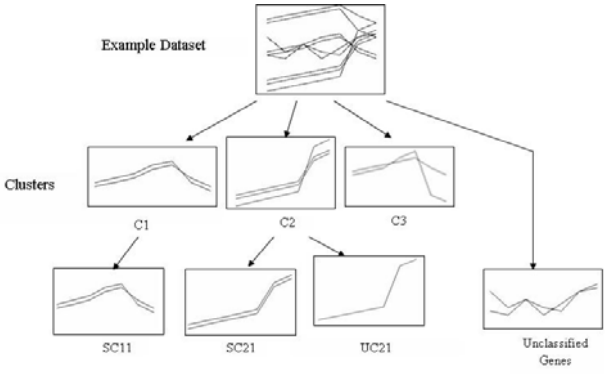

Fig. 3. Clustering of the example dataset given in Fig. 2. Here, C_is (i =1, 2, $\cdots$) are clusters; SC_{ij} refer to the j^{th} sub-cluster of cluster i and UC_{ik} is the k^{th} gene in cluster i not belonging to any sub-clusters.

The clusters and sub-clusters for the example dataset of Fig. 2 are illustrated in Fig. 3. It can be observed that sub-clusters give the highly coherent patterns in the dataset. The algorithms for cluster formation and cluster expansion are given in Fig. 4 and Fig. 5.

```
Cluster_creation()
//Pre-condition: All genes are unclassified
// cluster id = 0
FOR i from 1 to DG do
        IF gi.classified = unclassified THEN
                Cluster expand(gi , cluster id)
                cluster id++;
        END IF
END FOR
```

Fig. 4. Algorithm for cluster formation of GenClus

Microarrays generate tens of thousands of data in one experiment. Data volume is constantly increasing due to the huge amount of microarray experiments performed. While clustering this type of data, it is of utmost importance that the updations of the database are handled incrementally. Some of the incremental clustering algorithms are reported in section 2. Though a lot of work has focused on incremental clustering over spatial datasets, not much research has been done over incrementally handling gene expression data. In this paper, we also introduce an incremental clustering technique (InGenClus) for gene expression data, which is based on GenClus. Once clustering of the dataset is obtained, each of the clusters are represented by cluster profiles. The cluster profiles store the regulation of that particular cluster. The sub-clusters are represented by the sub-cluster profiles which stores the regulation and range information of that particular sub-cluster. This information is further used by GenClus when clustering the updated database incrementally.

Cluster_expand(g_i, cluster_id)
IF get_core(g_i) = 0 THEN
    g_i.cluster_id = cluster_id;
RETURN;
ELSE
    g_i.classified = classified;
    FOR j from 1 to D_G do
        IF g_j .classified = unclassified
            Cluster_expand(g_j , cluster_id)
        END IF
    END FOR
END IF

Fig. 5. Algorithm for cluster expansion of GenClus

4. Incremental Clustering

In this section, we present InGenClus which is based on GenClus and is capable of handling incremental data. Due to the density based nature of GenClus, the insertion of a gene affects the current clustering only in the neighborhood of the gene. We examine the parts of an existing clustering affected by an update and show how GenClus can handle incremental updates of a clustering after insertions. The changes of the clustering of the gene database D_G are restricted to the neighborhood of an inserted gene. The previously core genes [15] retain their core property but, non core genes (border genes or noise genes) may become cores. Thus new density connections may surface. The insertion of a gene g_i may result in a change of cluster membership of genes in the neighborhood of g_i and all genes reachable from one of these genes in $D_G^{'} = D_G \bigcup \{g_i\}$, where $D_G^{'}$ is the updated dataset. While inserting g_i the following cases may occur:

1. **Fusion**: A gene g_i may be fused to a cluster C_i if regulation pattern of g_i matches with cluster profile of C_i, then g_i is fused into cluster C_i. Gene g_i may be fused to a cluster S_i if g_i is reachable from S_i.
2. **Cluster Creation**: Gene g_i may have same regulation pattern w.r.t. some other noise or unclassified gene(s) and may lead to the formation of a new cluster.
3. **Sub-cluster Creation**: Gene g_i may become core w.r.t. (i) Some gene(s) in a cluster which are not members of any sub-cluster. This leads to the creation of a new sub-cluster. (ii) Some other noise or unclassified gene(s) and may lead to the formation of a new sub-cluster.
4. Noise: If g_i does not match with any of the cluster profiles then g_i is a noise gene and no density-connections are changed.

InGenClus starts with a newly inserted gene g_i and finds if its regulation and range information matches with any of the cluster or sub-cluster profiles then there can be the following cases:

i. g_i matches with cluster profile of C_i, then GenClus will assign cluster_id of C_i to g_i. After insertion of g_i, one of the genes $g_k \in C_i$ and $g_k \in S_i$ (S_i is a sub-cluster in C_i) may become core and hence can become a potential candidate for sub-cluster expansion (case 1).
ii. g_i matches with none of the cluster profiles, but it matches with some other unclassified genes. Then it creates a new cluster (case 2) and finds if it can form sub-clusters (case 3). Finally, it forms the cluster and/or sub-cluster profiles accordingly.
iii. g_i matches with cluster profile of C_i, then InGenClus will assign cluster_id of C_i to g_i. After insertion of g_i, any gene $g_k \in C_i$ and g_k not belonging to any sub-clusters in C_i may become core and hence may become a potential candidate for sub-cluster creation (case 3).
iv. g_i matches with none of the cluster profiles nor does it match with any other gene, then case 4 occurs.

In case of fusion, the affected cluster profiles are updated based on an effective data fusion technique. To achieve better space time complexity, the cluster profiles are organized using an effective data structure. It has been found that the InGenClus yields the same result as when compared with GenClus, yet at a lesser time.

5. Performance Evaluation

GenClus was implemented in Java in Windows environment and evaluated with several real-life datasets. GenClus was tested on the following three real-life data sets given in Table 1. All the datasets are normalized to have mean zero and standard deviation one. Of the various datasets, some of the clusters formed from the full and reduced form of Dataset 2 are shown in Fig. 6 and Fig.7 and the clusters obtained from the reduced Dataset 1 are shown in Fig.8. The datasets have been reduced by filtering out low variance genes and genes having more than 3-fold standard deviation. Some of the clusters obtained by GenClus from Dataset 3 are shown in Fig. 9. The hierarchy of four of the clusters and sub-clusters of

Table 1: Datasets used for evaluating the clustering algorithm

Serial No	*Dataset*	No. of genes	No. of conditions	Source
Dataset 1	Yeast CDC28-13 [33]	6218	17	http://yscdp.stanford.edu/yeast_cell_cycle/full_data.html
Dataset 2	Yeast Diauxic Shift [34]	6089	7	http://www.ncbi.nlm.nih.gov/geo/query
Dataset 3	Subset of Human Fibroblasts Serum [35]	517	13	http://www.sciencemag.org/feature/data/984559.hsl

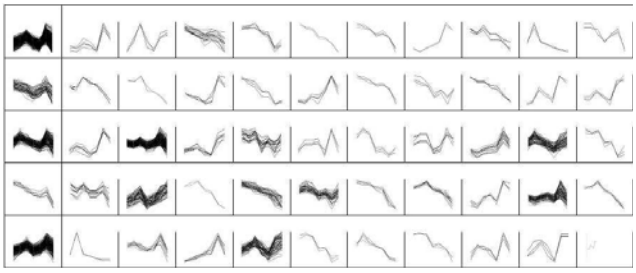

Fig. 6. Some clusters are illustrated from the full Dataset 2

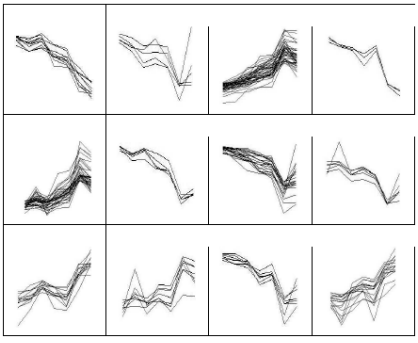

Fig. 7. Result of GenClus on the reduced form of Dataset 2

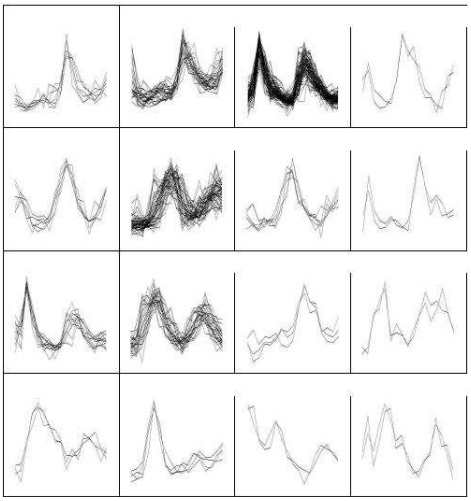

Fig. 8. Some clusters are illustrated from the Dataset 1

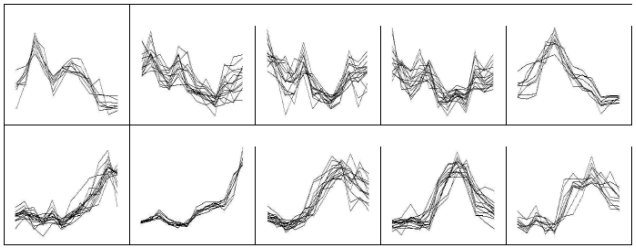

Fig. 9. Some of the clusters obtained by GenClus over Dataset 3

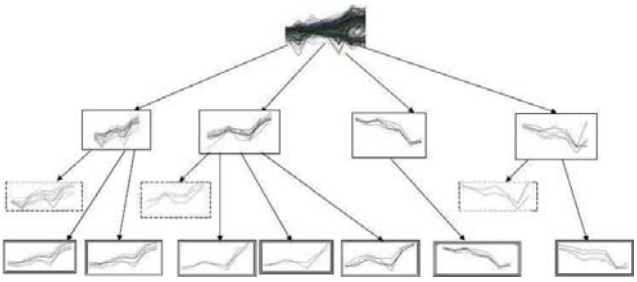

Fig. 10. Hierarchy of four clusters of Dataset 2. The full dataset is at the root, the clusters are shown with the single line frames, sub-clusters are shown with double line frames and the genes which are part of a higher level cluster but not part of any sub-clusters are shown with dotted line frames.

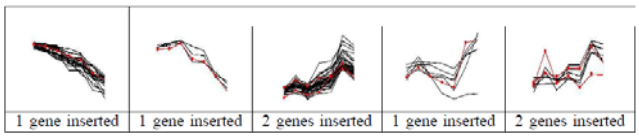

Fig. 11. Some of the clusters obtained by InGenClus over data incrementally updated from Dataset 2

Dataset 2 is shown in Fig. 10. In the figure, the full dataset is at the root, the clusters are shown with the single line frames, sub-clusters are shown with double line frames and the genes which are part of a higher level cluster but not part of any sub-clusters are shown with dotted line frames. The data from Dataset 2 was inserted incrementally and InGenClus was executed. Fig. 11 shows a sample output of some clusters of Dataset 2 with genes inserted incrementally. The inserted genes are shown in red color (grey for black & white images) with filled circles at the time points.

5.1. Cluster Quality

To assess the quality of our method, we need an objective external criterion. In order to validate our clustering result, we employed z-score and *p-value*.

Z-score*:* For evaluating the quality of clusters produced by different algorithms, we need an objective external criterion. We obtain a statistical rating of the relative gene expression activity shown by the genes associated in each cluster and the GO terms. In order to validate our clustering result, we employ z-score [36] as the measure of agreement. To assess the quality of GenClus, we employed z-score [36] as the measure of agreement. Higher the value of z, better the cluster results indicating more biologically relevant clusters of genes. z-score is calculated by investigating the relation between a clustering result and the functional annotation of the genes in the cluster. We have used Gibbons ClusterJudge [36] tool to calculate the z-score.

Table 2: z-scores for GenClus and its counterparts for Dataset 2

Method Applied	**No. of Clusters**	**Total no. of genes**	**z-score**
k-means	62	614	5.57
SOM	42	614	5.78
GenClus	61	614	7.39

To test the performance of the clustering algorithm, we compared the clusters identified by our method with the results from k-means and SOM. The result of applying the z-score on the reduced form of Dataset 2 is shown in Table 2. In this table GenClus was compared with the well known k-means and SOM. Similarly, InGenClus was implemented and tested over various datasets. The results were compared with GenClus and have been found satisfactory. Some of the results obtained by InGenClus over Dataset 2 are reported in Fig. 11. It has been found that InGenClus yields the same result as GenClus as can be observed from Table 3.

Table 3: z-scores for GenClus and In GenClus for Dataset 1

Method Applied	**No. of Clusters**	**Total no. of genes**	**z-score**
GenClus	21	384	11.68
InGenClus	21	384	11.68

Biological significance*:* The biological relevance of a cluster can be verified based on the gene ontology (GO) annotation database located at *http://db. yeastgenome.org /cgibin/GO/goTermFinder*. It is used to test the functional enrichment of a group of genes in terms of three structured controlled ontologies, *viz.*, associated biological processes, molecular functions and biological components. The functional enrichment of each GO category in each of the clusters obtained is calculated by its *p-value*. *p-value* represents the probability of observing the number of genes from a specific GO functional category within each cluster. A low *p-value* indicates the genes belonging to the enriched functional categories are biologically significant in the corresponding clusters. To compute the *p-value*, we used the software FuncAssociate [37]. FuncAssociate [37] computes the hypergeometric functional enrichment score based on Molecular Function and Biological Process annotations. The resulting scores are adjusted for multiple hypothesis testing using Monte Carlo simulations. FuncAssociate is a Web-based tool that accepts as input a list of genes and returns a list of GO attributes that are over-represented (or under-represented) among the genes in the input list.

To test the biological significance of the clusters obtained by GCA, we use a reduced form of Dataset 1 obtained from *http://faculty.washington.edu/kayee/cluster*. Functional categories with p-value$<7\times10^{-07}$ are reported in order to restrict the size of the paper. Of the various clusters obtained from the reduced form of Dataset 1, the highly enriched categories are shown in Table 4. As can be seen in cluster C3, the highly enriched categories of 'cell cycle', 'DNA metabolic process', 'DNA replication', 'chromosome', etc. have p-values of 1.1×10^{-25}, 7.6×10^{-22}, 4.1×10^{-21}, 3.4×10^{-18} respectively. The highly enriched categories in cluster C6 are the 'cellular bud' and the 'cell cycle' with *p-values* of 3×10^{-12} and 8.4×10^{-12} respectively. The genes in clusters C3 and C6 are involved in cell cycle. Cluster C9 have genes involved in DNA replication. The cluster C10 contains highly enriched categories such as 'spindle', 'cytoskeletal part', 'microtubule cytoskeleton' with *p-values* of 1.3×10^{-12}, 1.8×10^{-11}, 1.9×10^{-11} respectively. C10 contains functions related to various phases of cell cycle. Cluster C13 contains various functional categories related to synthesis of various amino acids. From the results of Table 4, we see that the clusters

obtained by GenClus shows a high enrichment of functional categories.

6. Conclusions

This work presents a density based clustering approach which finds useful subgroups of highly coherent genes within a cluster and obtains a hierarchical structure of the dataset where the sub-clusters give the finer clustering of the dataset. GenClus does not require the number of clusters apriori and the clusters obtained have been found satisfactory on visual inspection and also based on z-score for three real-life datasets. However, work is going on for establishing the effectiveness of GenClus over more real-life datasets. An incremental version of GenClus, i.e., InGenClus is also introduced in this paper which helps to update the clustering result of GenClus for incremental data. The algorithm brings down the cost of performing GenClus on the whole database after insertions are carried out. The number of neighborhood queries is scaled down much effectively than allowing it to run on the whole updated dataset at a time. InGenClus has been found faster, yet yielding the same result when compared with GenClus.

References

[1] D. Stekel, Microarray Bioinformatics. Cambridge, UK: Cambridge University Press, 2006.
[2] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," Journal of Computational Biology, vol. 6(3-4), pp. 281–297, 1999.
[3] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," in Proceedings of National Academy of Sciences, vol. 95, 1998, pp. 14 863–14 868.
[4] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," Available： www.cse.buffalo.edu/ DBGROUP /bioinformatics/papers/survey.pdf., 2003.
[5] D. Jiang, J. Pei, and A. Zhang, "DHC: a density-based hierarchical clustering method for time series gene expression data," in Proceedings of BIBE2003: 3rd IEEE International Symposium on Bioinformatics and Bioengineering, Bethesda, Maryland, 2003, p. 393.
[6] J. McQueen, "Some methods for classification and analysis of multivariate observations," in Proceedings of the Fifth Berkeley Symp. Math. Statistics and Probability, vol. 1, 1967, pp. 281–297.
[7] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," in Proceedings of National Academy of Sciences, vol. 96(6), USA, 1999, pp. 2907–2912.
[8] J. Dopazo and J. Carazo, "Phylogenetic reconstruction using an unsupervised neural network that adopts the topology of a phylogenetic tree," J Mol Eval, vol. 44, pp. 226–233, 1997.
[9] S. Tomida, T. Hanai, H. Honda, and T. Kobayashi, "Analysis of expression profile using fuzzy adaptive resonance theory," Bioinformatics, vol. 18(8), pp. 1073–83, 2002.
[10] U. Alon, N. Barkai, Notterman, D. A., K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array," in Proceedings of Natl. Acad. Sci, vol. 96(12), USA, 1999, pp. 6745–6750.
[11] A. Bhattacharya and R. De, "Divisive correlation clustering algorithm (DCCA) for grouping of genes: detecting varying patterns in expression profiles," Bioinformatics, vol. 24(11), pp. 1359–1366, 2008.
[12] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of International Conference on Knowledge Discovery in Databases and Data Mining (KDD-96), Portland, Oregon, 1996, pp. 226–231.

Table 4: *p-values* for some of the clusters of Dataset 1

Cluster	P-value	GO number	GO category
C3	1.1e-25	GO:0007049	cell cycle
	7.6e-22	GO:0006259	DNA metabolic process
	4.1e-21	GO:0006260	DNA replication
	3.4e-18	GO:0005694	chromosome
	2.2e-17	GO:0044427	chromosomal part
	5.6e-17	GO:0006261	DNA-dependent DNA replication
	1e-16	GO:0022402	cell cycle process
	3.6e-15	GO:0022403	cell cycle phase
	8.7e-15	GO:0006281	DNA repair
	1.5e-14	GO:0000278	mitotic cell cycle
	2.6e-14	GO:0005657	replication fork
	8.6e-14	GO:0006974	response to DNA damage stimulus
	2.5e-13	GO:0000228	nuclear chromosome
	3.1e-13	GO:0009719	response to endogenous stimulus
	1.9e-12	GO:0045934	negative regulation of nucleobase, nucleoside, nucleotide and nucleic acid metabolic process
	1.9e-11	GO:0051276	chromosome organization and biogenesis
	2.1e-11	GO:0006273	lagging strand elongation
	2.8e-11	GO:0044454	nuclear chromosome part
	4e-11	GO:0031324	negative regulation of cellular metabolic process
	4.3e-11	GO:0009892	negative regulation of metabolic process
	5.3e-11	GO:0005634	nucleus
	7.6e-11	GO:0051301	cell division
	8.5e-11	GO:0030894	replisome
	8.5e-11	GO:0043601	nuclear replisome
	1.1e-10	GO:0007064	mitotic sister chromatid cohesion
	1.3e-10	GO:0048523	negative regulation of cellular process
	1.7e-10	GO:0048519	negative regulation of biological process
	3e-10	GO:0006271	DNA strand elongation during DNA replication
	3e-10	GO:0022616	DNA strand elongation
	1.1e-09	GO:0006323	DNA packaging
	1.5e-09	GO:0000079	regulation of cyclic dependent protein kinase activity
	1.7e-09	GO:0005933	cellular bud
	2.1e-09	GO:0006342	chromatin silencing
	2.1e-09	GO:0031507	heterochromatin formation
	2.1e-09	GO:0045814	negative regulation of gene expression, epigenetic
	2.5e-09	GO:0043596	nuclear replication fork
	2.5e-09	GO:0005935	cellular bud neck
	3.8e-09	GO:0007062	sister chromatid cohesion
	4.2e-09	GO:0016458	gene silencing
	4.2e-09	GO:0040029	regulation of gene expression, epigenetic
	5.8e-09	GO:0051325	interphase
	6.1e-09	GO:0051726	regulation of cell cycle
	1.7e-08	GO:0003677	DNA binding
	2.7e-08	GO:0031497	chromatin assembly
	3.1e-08	GO:0000819	sister chromatid segregation
	3.4e-08	GO:0051052	regulation of DNA metabolic process
	3.6e-08	GO:0000279	M phase
	4.2e-08	GO:0045892	attribute negative regulation of transcription, DNA-dependent
	4.3e-08	GO:0051329	interphase of mitotic cell cycle
	4.5e-08	GO:0006139	nucleobase, nucleoside, nucleotide and nucleic acid metabolic process
	4.6e-08	GO:0051253	negative regulation of RNA metabolic process
	7.9e-08	GO:0030427	site of polarized growth
	8.9e-08	GO:0006333	chromatin assembly or disassembly
	1e-07	GO:0016481	negative regulation of transcription
	1e-07	GO:0043228	non-membrane-bounded organelle
	1e-07	GO:0043232	intracellular non-membrane-bounded organelle
	1.1e-07	GO:0043549	regulation of kinase activity
	1.1e-07	GO:0045859	regulation of protein kinase activity
	1.1e-07	GO:0000793	condensed chromosome
	1.3e-07	GO:0006310	DNA recombination
	1.3e-07	GO:0000731	DNA synthesis during DNA repair
	1.4e-07	GO:0051338	regulation of transferase activity
	1.5e-07	GO:0050794	regulation of cellular process
	3.7e-07	GO:0006950	cresponse to stress
	5.3e-07	GO:0006298	mismatch pair
	5.3e-07	GO:0045005	maintenance of fidelity during DNA-dependent DNA replication
	6.4e-07	GO:0000798	nuclear cohesin complex
	6.4e-07	GO:0008278	cohesin complex

Cluster	P-value	GO number	GO category
C6	3e-12	GO:0005933	cellular bud
	8.4e-12	GO:0007049	cell cycle
	5.3e-11	GO:0005935	cellular bud neck
	5.5e-11	GO:0030427	site of polarized growth
	1.2e-08	GO:0051301	cell division
	5.7e-08	GO:0022402	cell cycle process
	1.7e-07	GO:0000278	mitotic cell cycle
	6.1e-07	GO:0022403	cell cycle phase
C9	6.7e-09	GO:0042555	MCM complex
	6.5e-07	GO:0005656	pre-replicative complex
	6.8e-07	GO:0006267	pre-replicative complex assembly
C10	1.3e-12	GO:0005819	spindle
	1.8e-11	GO:0044430	cytoskeletal part
	1.9e-11	GO:0015630	microtubule cytoskeleton
	8.9e-11	GO:0005856	cytoskeleton
	3.6e-10	GO:0000278	mitotic cell cycle
	4e-10	GO:0007020	microtubule nucleation
	7.2e-10	GO:0007049	cell cycle
	7.7e-10	GO:0007017	microtubule-based process
	6.8e-10	GO:0008622	epsilon DNA polymerase complex
	2.1e-09	GO:0005200	structural constituent of cytoskeleton
	3.7e-09	GO:0000226	microtubule cytoskeleton organization and biogenesis
	4.2e-09	GO:0005874	microtubule
	5.3e-09	GO:0007059	chromosome segregation
	1.7e-08	GO:0000775	chromosome, pericentric region
	2.5e-08	GO:0043228	non-membrane-bounded organelle
	2.5e-08	GO:0043232	intracellular non-membrane-bounded organelle
	4.4e-08	GO:0007010	cytoskeleton organization and biogenesis
	4.5e-08	GO:0022402	cell cycle process
	5e-08	GO:0051301	cell division
	1.3e-07	GO:0000776	kinetochore
	1.4e-07	GO:0005815	microtubule organizing center
	1.4e-07	GO:0005816	spindle pole body
	2.2e-07	GO:0000070	mitotic sister chromatid segregation
	2.2e-07	GO:0005694	chromosome
	2.4e-07	GO:0000922	spindle pole
	3.6e-07	GO:0005822	inner plaque of spindle pole body
	3.9e-07	GO:0000819	sister chromatid segregation
	6.3e-07	GO:0044427	chromosomal part
C13	4e-07	GO:0009086	methionine biosynthetic process
	6.2e-07	GO:0000097	sulfur amino acid biosynthetic process

[13] G. Shu, B. Zeng, Y. Chen, and O. Smith, "Performance assessment of kernel density clustering for gene expression prole data," Comparative and Functional Genomics, vol. 4, p. 287299., 2003.
[14] R. Jarvis and E. Patrick, "Clustering using a similarity measure based on shared nearest neighbors," IEEE Transactions on Computers, vol. 11, 1973.
[15] R. Das, D. Bhattacharyya, and J. Kalita, "Clustering gene expression data using an effective dissimilarity measure," International Journal of Computational BioScience (Special Issue), vol. 1(1), pp. 55–68, 2010.
[16] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: The order-preserving submatrix problem," in Proc. Of the 6th Annual Int. Conf. on Computational Biology. New York, USA: ACM Press, 2002, pp. 49–57.
[17] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," J. R. Statist. Soc., vol. B 39 (1), pp. 1–38, 1977.
[18] J. Travis and Y. Huang, "Clustering of gene expression data based on shape similarity," EURASIP Journal on Bioinformatics and Systems Biology,, vol. 2009(195712), 2009.
[19] M. Beal and Z. Ghahramani, "The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures," in Proceedings of the 7th Valencia International Meeting on Bayesian Statistics, vol. 63(4), Spain, 2003, pp. 453–464.
[20] E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach, and R. Shamir, "An algorithm for clustering cDNAs for gene expression analysis using short oligonucleotide fingerprints," in Proceedings of 3rd International Symposium on Computational Molecular Biology (RECOMB 99). ACM Press, 1999, pp. 188–197.
[21] R. Sharan and R. Shamir, "CLICK: A clustering algorithm with applications to gene expression analysis," in Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology. AAAI Press, 2000.
[22] A. Bellaachia, D. Portnoy, and A. G. Chen, Y.and Elkahloun, "E-CAST: A data mining algorithm for gene expression data," in Proceedings of the BIOKDD02: Workshop on Data Mining in Bioinformatics (with SIGKDD02 Conference), 2002, pp. 49.
[23] R. Das, D. Bhattacharyya, and J. Kalita, "A new approach for clustering gene expression time series data," International Journal of Bioinformatics Research and Applications, vol. 5(3), pp. 310–328, 2009.
[24] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum Press, 1981.
[25] S. Bandyopadhyay, A. Mukhopadhyay, and U. Maulik, "An improved algorithm for clustering gene expression data," Bioinformatics, vol. 23(21), pp. 2859–2865, 2007.
[26] U. Maulik, A. Mukhopadhyay, and S. Bandyopadhyay, "Combining pareto-optimal clusters using supervised learning for identifying coexpressed genes," BMC Bioinformatics, vol. 10(27), 2009.
[27] M. Ester, H. P. Kriegel, J. Sander, M. Wimmer, and X. Xu, "An incremental clustering for mining in a data warehousing environment," in Proceedings of the 24th VLDB Conference, New York, USA, 1998.
[28] L. Tao and S. Sarabjot, "Hirel: An incremental clustering algorithm for relational datasets," in Proceedings of Eighth IEEE International Conference on Data Mining, 2008, pp. 887–892.
[29] S. Asharaf, M. Narasimha, and S. Shevade, "Rough set based incremental clustering of interval data," Pattern Recognition Letters, vol. 27, pp. 515–519, 2006.
[30] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. Brown, "Incremental genetic k-means algorithm and its application in gene expression data analysis," BMC Bioinformatics, vol. 5(172), 2004.
[31] R. Ruiz, J. Riquelme, and J. Aguilar-Ruiz, "Incremental wrapper-based gene selection from microarray data for cancer classification," Pattern Recognition, vol. 39, p. 23832392, 2006.
[32] R. Das, D. Bhattacharyya, and J. Kalita, "An incremental clustering of gene expression data," in Proceedings of NABIC, Coimbatore, India, 2009.
[33] R. J. Cho, M. Campbell, E. Winzeler, L. Steinmetz et al., "A genomewide transcriptional analysis of the mitotic cell cycle," Mol. Cell, vol. 2(1), p. 6573, 1998.
[34] J. DeRisi and P. Iyer, V.R.and Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," Science, vol. 278, pp. 680–686, 1997.
[35] V. Iyer, M. Eisen, D. Ross, G. Schuler, T. Moore, J. Lee, J. Trent, L. Staudt, J. Hudson, M. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P. Brown, "The transcriptional program in the response of the human fibroblasts to serum," Science, vol. 283, pp. 83–87, 1999.
[36] F. Gibbons and F. Roth, "Judging the quality of gene expression based clustering methods using gene annotation," Genome Research, vol. 12, pp. 1574–1581, 2002.
[37] F. G. Berriz et al., "Characterizing gene sets with funcassociate," Bioinformatics, vol. 19, pp. 2502–2504, 2003.

Sauravjyoti Sarmah is a Systems Analyst in the Department of Computer Science and Engineering, Jorhat Engineering College,

Jorhat, Assam, India. He is currently pursuing his Ph.D. in Computer Science at Tezpur University, Tezpur, India. His research interests include clustering and image processing. He has published several papers in international journals and referred conference proceedings.

Dhruba K. Bhattacharyya is a professor in the Department of Computer Science and Engineering, Tezpur University, Tezpur, India. He received his Ph.D. from Tezpur University in the year 1999. His research interests include data mining, network security and content-based image retrieval. He has published more than 100 papers in international journals and referred conference proceedings and has edited two books.