**IOSR Journal of Engineering (IOSR JEN)** **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 43-50

# Unsupervised Gene Selection Using EAGMFI

Dr.T.Chandrasekhar[1], Dr. D. Arul Pon Daniel[2]
[1]*Assistant Professor, Department of Computer Science, Asan College of Arts and Science, Karur, Tamil nadu, India*
[2]*Assistant professor, Department of Computer Applications, Loyola College of Arts and Science, Mettala, Namakkal, Tamil nadu, india*

***Abstract:*** *Microarrays are made it possible to simultaneously monitor the expression profiles of thousands of genes under various experimental conditions. It is used to identify the co-expressed genes in specific cells or tissues that are actively used to make proteins. This method is used to analysis the gene expression, an important task in bioinformatics research. Cluster analysis of gene expression data has proved to be a useful tool for identifying co-expressed genes, biologically relevant groupings of genes and samples. In this work the unsupervised Gene selection methods and Enhanced Automatic Generation of Merge Factor for ISODATA (EAGMFI) algorithms have been applied for clustering of Gene Expression Data. This proposed clustering algorithm overcomes the drawbacks in terms of specifying the optimal number of clusters and initialization of good cluster centroids. Gene Expression Data could identify compact clusters and best genes are selected with high performance is well in terms of the Silhouette Coefficients cluster measure.*

***Keywords:*** *Clustering, Gene expression data, Gene filter, USRR, EAGMF*

## I. Introduction

Clustering has been used in a number of applications such as engineering, biology, medicine and data mining. Cluster analysis of gene expression data has proved to be a useful tool for identifying co-expressed genes. DNA microarrays are emerged as the leading technology to measure gene expression levels primarily, because of their high throughput. Results from these experiments are usually presented in the form of a data matrix in which rows represent genes and columns represent conditions or samples [8]. Each entry in the matrix is a measure of the expression level of a particular gene under a specific condition. Analysis of these data sets reveals genes of unknown functions and the discovery of functional relationships between genes [10]. Co-expressed genes can be grouped into clusters based on their expression patterns of gene based clustering and Sample based clustering. 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 [11]. Discriminant analysis is now widely used in bioinformatics, such as distinguishing cancer tissues from normal tissues or one cancer subtype vs. another [3]. A critical issue in discriminant analysis is feature selection: instead of using all available variables (features or attributes) in the data, one selectively chooses a subset of features to be used in the discriminant system. There are a number of advantages of feature selection: (1) dimension reduction to reduce the computational cost; (2) reduction of noise to improve the classification accuracy; (3) more interpretable features or characteristics that can help identify and monitor the target diseases or function types. These advantages are typified in DNA microarray gene expression profiles. Of the tens of thousands of genes in experiments, only a smaller number of them show strong correlation with the targeted phenotypes [4]. In Karteeka Pavan et al proposed AGMFI Algorithm (Automatic Generation of Merge Factor for ISODATA) [7]. This paper studies an initialization of centroids proposed in for microarray data to get the best quality of clusters. A comparative analysis is performed for UCI data sets in-order to get the best clustering algorithm. Iterative Self-Organizing Data Analysis Techniques (ISODATA) tries to find the best cluster centres through iterative approach, until some convergence criteria are met. One significant feature of ISODATA over K-Means is that the initial number of clusters may be merged or split, and so the final number of clusters may be different from the number of clusters specified as part of the input. The ISODATA requires number of clusters, and a number of additional user-supplied parameters as inputs. To get better results user need to initialize these parameters with appropriate values by analyzing the input microarray data. The main difference between AGMFI and ISODATA is ISODATA uses heuristic values to merge the clusters, AGMFI generates automatically and the choice of centeroid (c) is not fixed but is to be decided to have better performance.

In the AGMFI algorithms also taken these K-Means clustering algorithms and it is very simple and fast efficient. Numerous methods have been proposed to solve clustering problem. The most popular clustering algorithms in microarray gene expression analysis are Hierarchical clustering, K-Means clustering [3], and SOM [9]. K-Means clustering algorithm which is developed by Mac Queen [5] and it is very effective in producing

clusters for many practical applications. But the computational complexity of the original K-Means algorithm is very high, especially for large data sets. The K-Means clustering algorithm is a partitioning clustering method that separates data into K groups. For the real life problems, the suitable number of clusters cannot be predicted. To overcome the above drawback the current research focused on developing the clustering algorithms without giving the initial number of clusters [1, 2, 4]. This paper is organized as follows. Section 2 presents pre processing techniques; Section 3 describes the gene filtering techniques. Section 4 presents unsupervised gene selection method; Section 5 describes the Unsupervised Clustering algorithm; Section 6 describes performance of Experimental analysis and discussion; Section 7 presents conclusion and future work.

## II. Pre Processiong Techniques

The purpose of clustering gene expression data is to reveal the natural structure inherent data and extracting useful information from noisy data. So, pre-processing of the data is an essential part in any data mining process. Some of the methods require the input data to be discrete, taken rough sets, clustering and association rules [3]. That is why a new task in pre-processing is needed as normalization and unsupervised discretization.

### *A. Z-Score Normalization*

The gene expression data is normalized to have mean 0 and standard deviation 1. Gene Expression Data having a low variance across conditions as well as Z-Score normalization methods [12] are used. The standard score is

$$z = \frac{x-\mu}{\sigma} \qquad (1)$$

Where: x is a raw score to be standardized, μ is the mean of the population, σ is the standard deviation of the population.

### *B. Discretization*

Many data mining techniques often require that the attributes of the data sets are discrete. Given that most of the experimental data are continuous, not discrete, the discretization of the continuous attributes is an important issue. The goal of discretization is to reduce the number of possible values a continuous attribute takes by partitioning them into a number of intervals. Discretization is then performed on this normalized expression data. The discretization is done as follows [12]

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

$$\xi_{g_i t_1} = \begin{cases} 1 \ if \ \varepsilon_{g_i t_j} > 0 \\ 0 \ if \ \varepsilon_{g_i t_1} = 0 \\ -1 \ if \ \varepsilon_{g_i t_1} < 0 \end{cases} \qquad (2)$$

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 – {$t_1$})

$$\xi_{g_i t_{j+1}} = \begin{cases} 1 \ if \ \varepsilon_{g_i t_j} < \varepsilon_{g_i t_{j+1}} \\ 0 \ if \ \varepsilon_{g_i t_j} = \varepsilon_{g_i t_{j+1}} \\ -1 \ if \ \varepsilon_{g_i t_j} > \varepsilon_{g_i t_{j+1}} \end{cases} \qquad (3)$$

here $\xi g_i$, $t_j$ is the discretized value of gene $g_i$ at conditions $t_j$ (j = 1, 2, ..., T − 1). The gene expression values are gene $g_i$ and condition $t_j$ is given by $g_i$, $t_j$. Compute first condition $t_1$, is treated as a special case and its discretized value is directly based on $g_i$, $t_1$ 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 of 0, 1, and -1 across the conditions or time points. This pattern is represented as a string.

## III. Gene Filtering Techniques

The Bioinformatics Toolbox™ product extends the MATLAB® environment to provide an integrated software environment for genome and proteome analysis. The Bioinformatics Toolbox product includes many functions to help us with genome and proteome analysis. Most functions are implemented in M-code (the MATLAB programming language) with the source available for us to view. One can use the basic bioinformatics functions provided with this toolbox to create more complex algorithms and applications. These robust and well-tested functions are used to gene analysis.Gene expression profile experiments have data where the absolute values are very low. The quality of this type of data is often bad due to large quantization errors or

simply poor spot hybridization. We can use filtering functions to clean raw data before analysis with bellow gene filtering methods.

***$F_1$=geneentropyfilter(z)*** **-** This method removes genes with low entropy expression values

***$F_2$=generangefilter(z)*** **-** This method removes gene profiles with small profile ranges

***$F_3$=genevarfilter(z)*** **-** This method Filter gene with small profile variance

Here z - discretized values, $F_1$, $F_2$, $F_3$ – Filtered gene expression data.

## IV. Unsupervised Gene Selection

Gene selection is an important problem in the research of Gene expression data analysis. In some cases, too many redundant or missing values are there in gene expression data. In this section, an existing works of Velayutham and Thangavel et al. proposed an algorithm of unsupervised feature selection (Reduct algorithms) [14] is applied. This method is based on relative dependency measure using rough set theory.

Rough set theory can be regarded as a new mathematical tool for imperfect data analysis. The theory has found applications in many domains. Objects characterized by the same information are indiscernible (similar) in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory. Any set of all indiscernible (similar) objects is called an elementary set, and forms a basic granule (atom) of knowledge about the universe [16]. Any union of some elementary sets is referred to as a crisp (precise) set – otherwise the set is rough (imprecise, vague). Rough set is a pair of precise sets, called the lower and the upper approximation of the rough set, and is associated. The lower approximation consists of all objects which surely belong to the set and the upper approximation contains all objects which possibly belong to the set. The difference between the upper and the lower approximation constitutes the boundary region of the rough set. From a data table are called columns of which are labeled by attributes, rows – by objects of interest and entries of the table are attribute values. In data mining applications, decision class labels are often unknown or incomplete. In this situation the unsupervised feature selection is play vital role to select features.

*A. Relative Dependency Measure [14]*

The unsupervised relative dependency measure for an attribute subset is defined as follows

$$K_R(\{a\}) = \frac{\left|\frac{U}{IND(R)}\right|}{\left|\frac{U}{IND(R \cup \{a\})}\right|}, \forall a \in A \qquad (4)$$

where R is a reduct if and only if $K_R(\{a\}) = K_c(\{a\})$ and $\forall x \subset R,\ Kx(\{a\}) \neq Kc(\{a\})$

In this case, the decision attribute used in the supervised feature selection, is replaced by the conditional attribute which is to be eliminated from the current reduct set R. So the gene expression data can be changed as the following methods

**Table 1** Arrange All he Gene Expression Data Set

| Samples | Condition attributes(genes) | | | |
|---|---|---|---|---|
| | **Gene 1** | **Gene 2** | **Gene 3…** | **Gene n** |
| 1 | g(1,1) | g(1,2) | … | g(1,n) |
| 2 | g(2,1) | g(2,2) | … | g(2,n) |
| … | … | … | … | … |
| M | g(m,1) | g(m,2) | … | g(m, n) |

*B. Unsupervised Relative Reduct (USRR) Algorithm [14]*

The proposed USSR algorithm starts by considering all of the features contained in the dataset.

**Algorithm 1:** USRelativeReduct(C)

C, the set of all conditional attributes;

(a) R ← C

(b) ∀a ∈ C

(e) **if**( $K_{R-\{a\}}(\{a\}) = = 1$)

(f) R ← R-{a}

(g) **return** R

Each attribute is then examined iteratively, and the relative dependency measure is calculated. If the relative dependency is equal to 1 then that attributes can be remove. This process continues until all features have been examined.

## V. Unsupervised Clustering Algorithms

The reduced data selected by the gene selection algorithm USRR are clustered by using the EAGMFI. This Algorithm is our hybrid proposed work; it is used Enhanced initial centroid selection with AGMFI Algorithms. AGMFI is also merging the K-Means clustering algorithms [7]. The main work of AGMFI is merging and split the cluster automatically, but the drawback of choice of centroid (c values) is not fixed. If the choice is fixed; it may be have a better cluster performance. So we select our proposed work of EAGMFI Algorithms. In this method may take initial centroid selection automatically like as AGMFI Algorithm. The distance measure used here is the Euclidean distance. To assess the quality of the clusters, we used the silhouette measure proposed by Rousseeuw [9].

### *A. Enhanced Automatic Generation of Merge Factor for ISODATA (EAGMFI) Algorithm*

Performance of iterative clustering algorithms which converges to numerous local minima depends highly on initial cluster centers. Generally initial cluster centers are selected randomly. The cluster centre initialization algorithm is studied to improve the performance of the K-Means algorithm. The clusters produced in the Enhanced initial centroid with K-Means clustering are further optimized by ISODATA algorithm. Some of the parameters are fixed by user during the merging and partitioning the clusters. The step by step procedure of EAGMFI is given below [13].

**Algorithm 2**: The EAGMFI algorithm

**Require**: $D = \{d_1, d_2, d_3, ..., d_n\}$ // Set of n data points. K - Number of desired clusters.
$d_i = \{ x_1, x_2, x_3, ..., x_i, ..., x_m \}$ // Set of attributes of one data point.
m- minimum number of samples in a cluster. N – maximum number of iterations.
Θs – a threshold value for spilt_size., Θc - a threshold value for merge_size.

**Ensure**: A set of K clusters.

**Steps**:

1. In the given data set *D*, if the data points contain the both Positive and negative attribute values then go to Step 2, otherwise go to step 4.
2. Find the minimum attribute value in the given data set *D*.
3. For each data point attribute, subtract with the minimum attribute value.
4. For each data point calculate the distance from origin.
5. Sort the distances obtained in step 4. Sort the data points accordance with the distances.
6. Partition the sorted data points into k equal sets.
7. In each set, take the middle point as the initial centroid.
8. Compute the distance between each data point $d_i$ $(1 \le i \le n)$ to all the initial centroids $c_j$ $(1 \le j \le k)$.
9. **Repeat**
10. For each data point $d_i$, find the closest centroid $c_j$ and assign $d_i$ to cluster *j*.
11. Set ClusterId[*i*] =*j*. // *j*: Id of the closest cluster.
12. Set NearestDist[*i*] = $d(d_i, c_j)$.
13. For each cluster $j$ $(1 \le j \le k)$, recalculate the centroids.
14. **For** each data point $d_i$,

14.1 Compute its distance from the centroid of the present nearest cluster.
14.2 If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster. Else
14.2.1 For every centroid $c_j$ $(1 \le j \le k)$ compute the distance $d(d_i, c_j)$. **End for**;
**Until** the convergence criteria is met. (Identify clusters using Enhanced initial K-Means algorithms)
15. Find the inter distance in all other cluster to minimum average inter distances clusters point in C;
16. Discard the m and merging operations of cluster ≥ 2*K, If n is even go to step 4 or 5;
17.Distance between two centroids < C, merge the cluster and update centroid, otherwise repeat up to K/2 times;
18. K ≤ K/2 or n is odd go to step 6 or 7;
19. Find the standard division of all clusters that has exceeds S * standard division of D;
20. Executed N times or no changes occurred in clusters since the last time then stop, otherwise take the centroids of the clusters as new seed points and find the clusters using K- Means and go to step 3.

## VI. Experimental Analysis And Discussion

In this section, we describe the Microarray gene expression data sets used to analyze the methods studied in sections 4 only which are arranged for the listed in Table.1, number of features/genes are in column wise, and number of items/samples are in row wise. Otherwise interchange the row and column wise all the section 2, 3 and 5.

**Table 2:** Selected Genes From USRR Before Filtering Methods

| Data Set | Original Gene Data | USRR Selected Genes before_Filtering |
|---|---|---|
| Serum | 517*17 | 54*17 |
| Yeast | 2882*17 | 71*17 |
| Simulated | 6763*12 | 129*12 |
| Leukemia | 7129*34 | 157*34 |

***a. Serum data***

This data set is described and used in [7]. It can be downloaded from: http://www.sciencemag.org/ feature/ data/ 984559.shl and corresponds to the selection of 517*17 genes whose expression varies in response to serum concentration inhuman fibroblasts.

***b.Yeast data***

This data set is downloaded from Gene Expression Omnibus-databases. The Yeast cell cycle dataset contains 2884 genes and 17 conditions. To avoid distortion or biases arising from the presence of missing values in the data matrix we removal all the genes that had any missing value. This step results in a matrix of size 2882 * 17.

***c. Simulated data***

It is downloaded from http://www.ncbi.nlm.nih.gov/gds/. The set contains 6763*12 Genes [2].

*d. Leukemia data*

It is downloaded from the website: http://datam.i2r.a-star.edu.sg/datasets/krbd/. The set contains 7129*34.

**Table 3:** Selected Genes Using Filter Methods

| S.No | Data Set | Selected Genes from Filter method | | | Selected Genes from Filter with USRR method | | |
|---|---|---|---|---|---|---|---|
| | | $F_1$ | $F_2$ | $F_3$ | $F_1$ | $F_2$ | $F_3$ |
| 1 | Serum 517*17 | 465*17 | 461*17 | 465*17 | 43*17 | 45*17 | 46*17 |
| 2 | Yeast 2882*17 | 1193*17 | 1193*17 | 1193*17 | 56*17 | 59*17 | 63*17 |
| 3 | Simulated 6763*12 | 5964*12 | 6087*12 | 6087*12 | 97*12 | 113*12 | 132*12 |
| 4 | Leukemia 7129*34 | 6415*34 | 6416*34 | 6416*34 | 149*34 | 156*34 | 159*34 |

**Table 4:** EAGMFI Clustering for Serum Data

| Serum Data sets | USRR Before Filtering (54*17) | | Gene Entropy filter with USRR- $F_1$ (43*17) | | Gene Range filter with USRR-$F_2$ (45*17) | | Gene var filter with USRR-$F_3$ (46*17) | |
|---|---|---|---|---|---|---|---|---|
| | No. of Genes | Measure Value | No. of Genes | Measure Value | No. of Genes | Measure value | No. of Genes | Measure value |
| C1 | 7 | 0.460 | 6 | 0.371 | 5 | 0.471 | 4 | 0.411 |
| C2 | 6 | 0.451 | 4 | 0.582 | 8 | 0.382 | 9 | 0.318 |
| C3 | 8 | 0.270 | 3 | 0.473 | 3 | 0.273 | 5 | 0.331 |
| C4 | 9 | 0.155 | 5 | 0.282 | 7 | 0.182 | 7 | 0.182 |
| C5 | 7 | 0.189 | 13 | 0.360 | 12 | 0.216 | 8 | 0.326 |
| C6 | 7 | 0.160 | 7 | 0.379 | 6 | 0.211 | 9 | 0.211 |
| C7 | 10 | 0.067 | 5 | 0.332 | 4 | 0.112 | 4 | 0.132 |
| Total Gene | 54 | | 43 | | 45 | | 46 | |
| Avg | 0.250 | | 0.397 | | 0.264 | | 0.273 | |

**Table 5:** EAGMFI Clustering for Yeast Data

| Simulated Data sets | USRR Before Filtering (71*17) | | Gene Entropy filter with USRR- $F_1$ (56*17) | | Gene Range filter with USRR- $F_2$ (59*17) | | Gene var filter with USRR- $F_3$ (63*17) | |
|---|---|---|---|---|---|---|---|---|
| | No. of Genes | Measure Value | No. of Genes | Measure value | | No. of Genes | Measure value | No. of Genes |
| C1 | 9 | 0.241 | 9 | 0.571 | C1 | 9 | 0.241 | 9 |
| C2 | 8 | 0.278 | 4 | 0.322 | C2 | 8 | 0.278 | 4 |
| C3 | 12 | 0.314 | 11 | 0.373 | C3 | 12 | 0.314 | 11 |
| C4 | 10 | 0.291 | 7 | 0.425 | C4 | 10 | 0.291 | 7 |
| C5 | 11 | 0.482 | 3 | 0.326 | C5 | 11 | 0.482 | 3 |
| C6 | 13 | 0.325 | 12 | 0.411 | C6 | 13 | 0.325 | 12 |
| C7 | 8 | 0.281 | 10 | 0.422 | C7 | 8 | 0.281 | 10 |
| Total Gene | 71 | | 56 | | 59 | | 63 | |
| Avg | 0.316 | | 0.407 | | 0.195 | | 0.192 | |

**Table 6:** EAGMFI Clustering for Simulated Data

| Yeast Data sets | USRR Before Filtering (129*12) | | Gene Entropy filter with USRR- $F_1$ (97*12) | | Gene Range filter with USRR- $F_2$ (113*12) | | Gene var filter with USRR- $F_3$ (132*12) | |
|---|---|---|---|---|---|---|---|---|
| | No. of Genes | Measure Value | No. of Genes | Measure Value | | No. of Genes | Measure value | No. of Genes |
| C1 | 16 | 0.373 | 24 | 0.571 | C1 | 16 | 0.373 | 24 |
| C2 | 26 | 0.322 | 12 | 0.322 | C2 | 26 | 0.322 | 12 |
| C3 | 23 | 0.334 | 23 | 0.373 | C3 | 23 | 0.334 | 23 |
| C4 | 19 | 0.260 | 8 | 0.482 | C4 | 19 | 0.260 | 8 |
| C5 | 18 | 0.211 | 17 | 0.326 | C5 | 18 | 0.211 | 17 |
| C6 | 18 | 0.220 | 6 | 0.311 | C6 | 18 | 0.220 | 6 |
| C7 | 9 | 0.312 | 7 | 0.212 | C7 | 9 | 0.312 | 7 |
| Total Gene | 129 | | 97 | | 113 | | 132 | |
| Avg | 0.291 | | 0.371 | | 0.282 | | 0.307 | |

### *A. Comparative analysis of Gene selection*

The Microarray gene expression data sets after the section 2, uses to section 3 and 4 as gene filtering and unsupervised gene selection method the comparative analysis table 2 and 3. These selected genes are used to section 5.

**Table 7:** EAGMFI Clustering for Leukemia Data

| Leukemia Data sets | USRR Before Filtering (157*34) | | Gene Entropy filter with USRR-$F_1$ (149*34) | | Gene Range filter with USRR- $F_2$ (156*34) | | Gene var filter with USRR- $F_3$ (159*34) | |
|---|---|---|---|---|---|---|---|---|
| | No. of Genes | Measure Value | No. of Genes | Measure Value | | No. of Genes | Measure value | No. of Genes |
| C1 | 23 | 0.182 | 27 | 0.321 | C1 | 23 | 0.182 | 27 |
| C2 | 21 | 0.144 | 12 | 0.286 | C2 | 21 | 0.144 | 12 |
| C3 | 22 | 0.159 | 8 | 0.137 | C3 | 22 | 0.159 | 8 |
| C4 | 19 | 0.164 | 31 | 0.348 | C4 | 19 | 0.164 | 31 |
| C5 | 27 | 0.127 | 14 | 0.226 | C5 | 27 | 0.127 | 14 |
| C6 | 20 | 0.154 | 22 | 0.114 | C6 | 20 | 0.154 | 22 |
| C7 | 25 | 0.162 | 35 | 0.262 | C7 | 25 | 0.162 | 35 |
| Total Gene | 157 | | 149 | | 156 | | 159 | |
| Avg | 0.156 | | 0.242 | | 0.136 | | 0.178 | |

**Table 8:** EAGMFI Clustering Average Measure

| Data Sets | USRR Before Filtering | | Gene Entropy filter with USRR-$F_1$ | | Gene Range filter with USRR- $F_2$ | | Gene var filter with USRR- $F_3$ | |
|---|---|---|---|---|---|---|---|---|
| | Genes | Avg | Genes | Avg | Genes | Avg | Gene | Avg |
| Serum | 54 | 0.250 | 43 | 0.397 | 45 | 0.264 | 46 | 0.273 |
| Yeast | 71 | 0.316 | 56 | 0.407 | 59 | 0.195 | 63 | 0.192 |
| Simulated | 129 | 0.291 | 97 | 0.371 | 113 | 0.282 | 132 | 0.307 |
| Leukemia | 157 | 0.156 | 149 | 0.242 | 156 | 0.136 | 159 | 0.178 |

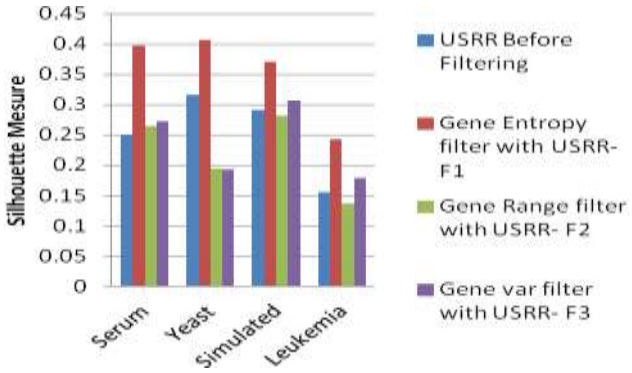


**Figure 1**: Performance Comparison of Silhouette Measure for Filter Methods

### *B. Comparative analysis of EAGMFI*

The EAGMFI used to cluster all data sets after the section 2, 3 and 4 is Table 2 and 3. In this EAGMFI method take initial value k as 7 then run as 10 times running to clusters data into 7 groups as comparative analysis Table 4, 5, 6 and 7.

Table 8 present the results of the FOUR different methods on the 4 data set. It shows the size of reduct found for each method, as well as the size of the optimal (minimal) reduct of data sets. In every sections we are reduct the gene expression data sets Figure 2, but in the second section we perform the pre-processing work only. From that the Gene Entropy filter with USRR-F1 method gives the optimum result Figure 1.

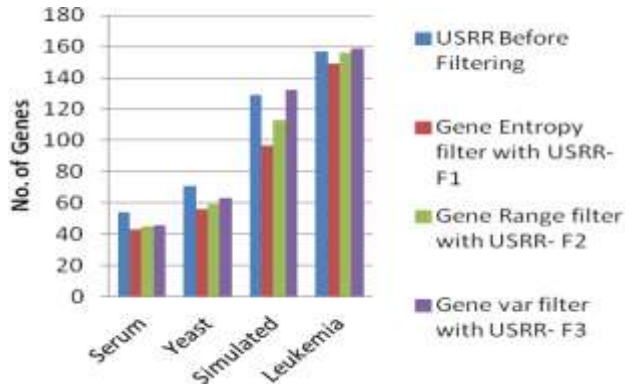


**Figure 2:** Comparison of Selected Genes using Filter Methods

## VII. Conclusion

In this work, Gene filtering techniques and unsupervised gene selections are studied to apply and avoid too many redundant or missing values in Microarray gene expression data. Both techniques are reduct the irrelevant data sets. In this unsupervised gene selection method is based on unsupervised Feature selection using Rough set method reduces gene data set. These methods are used to get minimum number of random gene data sets and then we use EAGMFI clustering technique to improve the quality of clusters. One of the demerits of AGMFI algorithm is random selection of initial seed point of desired clusters. This was overcome with EAGMFI for finding the initial centroids to avoid the random selection of initial values. Therefore, the EAGMFI algorithm is not depending upon any choice of the number of cluster and the evaluation of initial seed centroids will be selected automatically and it produces different optimum results. All the algorithms were tested with gene expression data and analysis the performance of cluster values using silhouette measurement. Therefore the findings give the solution to select the different centroids as a clusters seed points and various measures are used to improve the cluster performance is our future endeavor.

## References


[1]. Bashar Al-Shboul and Sung-Hyon Myaeng,(2009) "Initializing K-Means using Genetic Algorithms", World Academy of Science, Engineering and Technology 54.

[2]. Chen Zhang and Shixiong Xia,(2009) " K-Means Clustering Algorithm with Improved Initial center," in Second International Workshop on Knowledge Discovery and Data Mining (WKDD), pp. 790-792.

[3]. Chris Ding and Hanchuna Peng,(2003) "Mininmum Redundancy Feature Selection from Microarray Gene Expression Data", proceedings of the International Bioinformatic Conference, Date on(Aug) 11-14.

[4]. Dongxiao Zhu, Alfred O Hero, Hong Cheng, Ritu Khanna and Anand Swaroop,(2005) "Network constrained clustering for gene microarray Data", doi:10.1093 /bioinformatics / bti 655, Vol. 21 no. 21, pp. 4014 – 4020.

[5]. Doulaye Dembele and Philippe Kastner (2003), "Fuzzy C means method for clustering microarray data", Bioinformatics, vol.19, no.8, pp.973- 980.

[6]. F. Yuan, Z. H. Meng, H. X. Zhangz, C. R. Dong (2004), " A New Algorithm to Get the Initial Centroids", proceedings of the 3rdInternational Conference on Machine Learning and Cybernetics,(Aug) pp. 26-29.

[7]. K Karteeka Pavan, Allam Appa Rao, A V Dattatreya Rao, GR Sridhar,(2008). "Automatic Generation of Merge Factor for Clustering Microarray Data", IJCSNS International Journal of Computer Science and Network Security, (Sep),Vol.8, No.9.

[8]. K.R De and A. Bhattacharya,(2008). "Divisive Correlation Clustering Algorithm (DCCA) for grouping of genes: detecting varying Patterns in expression profiles," bioinformatics, Vol. 24, pp. 1359-1366.

[9]. Lletí, R., Ortiz, M.C., Sarabia, L.A., Sánchez, M.S,(2004). "Selecting variables for K-Means cluster analysis by using a genetic algorithm that optimises the silhouettes". Analytica Chimica Acta.

[10]. Sunnyvale, Schena M,(2000). "Microarray biochip technology". CA: Eaton Publishing;.

[11]. Sauravjoyti Sarmah and Dhruba K. Bhattacharyya(2010). "An Effective Technique for Clustering Incremental Gene Expression data", IJCSI International Journal of Computer Science Issues,(May) Vol. 7, Issue 3, No 3.

[12]. T.Chandrasekhar, K.Thangavel and E. Elayaraja,(2011). "Effective Clustering Algorithms for Gene Expression Data", IJCA International Journal of Computer Applications,(Oct), Volume 32, No- 4, ISSN: 0975-8887.

[13]. T.Chandrasekhar, K.Thangavel and E. Elayaraja,(2011). "Performance analysis of Enhanced Clustering Algorithm for Gene Expression Data ", IJCSI International Journal of Computer Science Issues, (Nov),Vol.8,Issue 6, No 3, ISSN (online):1694-0814.

[14]. Velayutham.C and Thangavel.K, (2011). "Unsupervised Feature Selection Using Rough Set", Proceeding on international Conference Emerging Trends in Computing, (Mar). 17-18.

[15]. Xiaosheng Wang, Osamu Gotoh,(2009). "Cancer Classification Using Single Genes", 20th international conference on Genome informatics(GIW 2009),14(Dec), vol.23, pp 179-188.

[16]. Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. Brown,(2004). "Incremental Genetic K-Means Algorithm and its Application in Gene Expression Data Analysis", BMC Bioinformatics.