Available online at: https://ijact.in

Date of Submission	26/06/2019
Date of Acceptance	03/09/2019
Date of Publication	05/10/2019
Page numbers	3422-3430(9 Pages)

Cite This Paper: Ameer A. AL-Mshanji, Sura Z.Al-Rashid. Improving Clustering Algorithm for GENE Expression Data Using Hybrid Algorithm, 8(9), COMPUSOFT, An International Journal of Advanced Computer Technology. PP. 3422-3430.

ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

IMPROVING CLUSTERING ALGORITHM FOR GENE EXPRESSION DATA USING HYBRID ALGORITHM

Ameer A. AL-Mshanji[1], Sura Z.AL-Rashid[2],
Software Department, College of Information Technology, University of Babylon, Hilla, Iraq
ameer.ali@uobabylon.edu.iq[1], sura_os@itnet.uobabylon.edu.iq[2]

Abstract: The technology of DNA Microarray has the ability to measure the levels of gene expression in different experimental conditions. Thousands of genes are generated in microarray experiments. The problem is that not all genes are significant; some of the genes may be noisy and irrelevant. The algorithms of Gene Selection are one of the important steps in the discovery of knowledge to select genes which are more informative. The other central goal of analyzing the data of gene expression is to identify genes that have similar patterns by using clustering processes. Clustering is a crucial process in the processes of data mining. It can divide genes into groups so that genes within the same group have similar features and share common biological functions. In this study, the method of mutual information for gene selection has been applied because it is able to detect nonlinear relationships between genes data. After that, the K-Means algorithm is applied to cluster data. The proposed approach results showed that it is capable of refining the data of gene expression for improved quality of clusters, handling noise effectively, and reducing the computational space.

Keywords: Microarray Technology; Gene Expression Data; Genes Selection; Clustering Algorithms; Clustering validation;

I. INTRODUCTION

With rapid technology development, Microarray Technology has become one of the most powerful tools in bioinformatics. Microarray technology is a good technique to observe the thousands of gene expression levels under different conditions at the same time. The conditions are usually consecutive time points during some environmental changes[1]. It can be beneficial to understand gene networks and functions in addition to its assistance in discovering the effects of medical treatments for diagnosing disease cases. The original gene data faces several problems such as missing values, noise and some variations. Therefore, the pre-processing of data is needed before any analysis [2]. The analysis of expression data can take two forms: it could either be a supervised analysis or an unsupervised one. In the supervised analysis, it is assumed that the structure data of the object is known. This knowledge is useful and can be applied in the analysis process. For the unsupervised analysis, the previously mentioned knowledge is not recognized.[3].Clustering (unsupervised) is a significant stage in the process of analyzing gene expression data and has been put into use in a wide range of fields such as medicine, biology, and engineering. Algorithms of clustering can able to discover the genes groups that exhibit similar expression patterns[4]. Clustering divides data points into sets or groups called clusters so that the homogeneity of elements in the same cluster shares some sort of strong similarity which is otherwise lower for the elements in other clusters. Clustering algorithms can cluster genes that have similar functions into clusters depending on the similarities of gene expression data, which help to understand the regulation of genes, processes of cellular, functions of genes, and the subtypes of cells[5]. Evaluating of clustering results is as important as the process of generating clusters. The validation of clustering techniques involves the possibility of providing an analytical assessment of structure type that has been captured by partitioning. Therefore, they should be an essential tool in interpreting the results of clustering. [6].

This paper is organized as follows: section 2 illustrates the related work. Section 3 presents a Microarray Technology overview. The gene expression matrix has been demonstrated in Section 3. Section 4 shows the used methods. The methodology is shown in Section 5. Section 6 presents the results and discussion. Finally, section 7 explains the conclusion.

II. RELATED WORK

Jacophine et al. (2015) used a hybrid system that applies agenetic algorithm and an adaptive pillar clustering algorithm. The proposed system was tested with three datasets of gene expression: thyroid, lung, and leukemia data. The hybrid system selects the optimal cluster among improved clusters[7]. Thomas et al. (2016) used Possibilistic Fuzzy C Means (PFCM) that represents a hybridization of Possibilistic C- Means (PCM) and Fuzzy C Means (FCM).They used lung data in their experiments. The experiments showed that the results are relatively better than PCM and FCM algorithms[8]. Angela et al. (2016) used different distance measures to optimize the performance of clustering algorithms. The experiments are implemented on Genomic Hybridization (CGH) dataset of cancer. The experiment results showed that using a distance matrix of gene expression data was an essential preprocessing step before using clustering algorithms [9]. Jorge et al. (2017) used a multi-objective evolutionary algorithm (MOEA).The proposed algorithm was implemented on three gene expression datasets: Arabidopsis thaliana, Medulloblastoma metastasis, and Yeast cell cycle. The results of the proposed model were gene clusters with higher levels of co-expression and biological functions than traditional single-objective clustering techniques [10]. Philip et al. (2018)used Pearson's Correlation Coefficient (PCC) to improve clustering tasks. In their experiments, they used gene expression data for the marine bacteria Crocosphaerawatsonii. The results of the proposed approach showed the biological clusters to be more reliable[11].Na Yuet al. (2018) used Non-negative Matrix Factorization (MvNMF) to discover co-differential genes. The suggested method has been examined in four multi genomic datasets: pancreatic adenocarcinoma, esophageal carcinoma, colon adenocarcinoma, neck squamous and head cell carcinoma data. The results of the method used presented a comparatively better performance than other methods [12].

III. MICROARRAY TECHNOLOGY

Microarray is a collection of thousands of DNA spots associated with a hard surface. The spots contain several duplications of the same DNA sequence that uniquely represents a gene from an organism. They are rigidly arranged and organized in the form of pen groups[13]. The expression level for each gene can be stored as an image (CEL file), and the data is extracted from the image by using special software. The surface of the DNA microarray is presented in Figure 1.

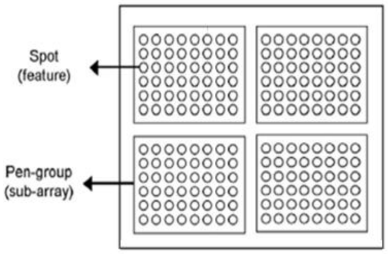


Figure 1: The surface of DNA microarray[13]

Most manufacturers of microarray provide their own software. The Puma package, for instance, is a suite of analysis methods for raw CEL file of microarray data [14].Scientists use DNA microarrays to measure the levels of gene expression for a huge number of genes simultaneously. This technology has helped scientists understand the basic aspects of life as well as explore genetic causes that are responsible for deviations in human body functions[15].

IV. GENE EXPRESSION MATRIX

The data is extracted from the microarray Image file using a special analysis program. This extracted data can be demonstrated in the shape of a matrix, often called the matrix of gene expression. It contains rows that express the genes and columns which express the special conditions at different times[16]. The collection of this data represents the basis for any analysis. Figure 2 shows the Gene expression matrix Structure.

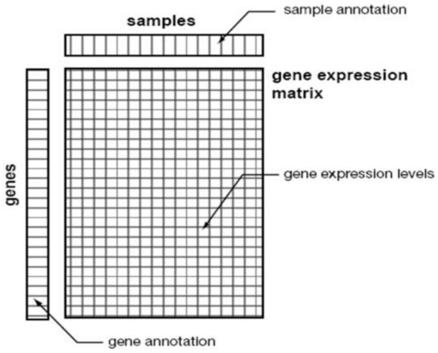


Figure 2: Gene expression matrix structure[17]

V. MATERIALS AND METHODS

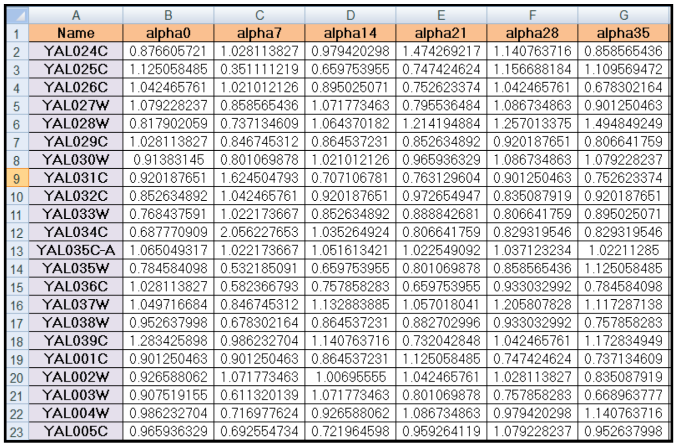
5.1 Dataset

In this study, the Yeast Cell Cycle gene expression data set has been used. It includes 6100 genes with 59 samples or conditions. Only two experiments have used alpha (contains 18 time series), cdc15 (contains 24 time series) and cdc28 (contains 17 time series). Table 1 summarizes the general information of the dataset. The dataset was downloaded fromhttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC25624/

Table 1: Summary of the Yeast Cell Cycle dataset

Title of Database :	Yeast Cell Cycle
Data Set Characteristics :	Multivariate
Attribute Characteristics :	String, Real
Missing Values?	Yes
Number of Instances:	6100
Number of Attributes:	60

Figure 3 depicts a screenshot for the file data in Excel format and Figure 4 illustrates the number of conditions in each experiment.



	A	B	C	D	E	F	G
1	Name	alpha0	alpha7	alpha14	alpha21	alpha28	alpha35
2	YAL024C	0.876605721	1.028113827	0.979420298	1.474269217	1.140763716	0.858565436
3	YAL025C	1.125058485	0.351111219	0.659753955	0.747424624	1.156688184	1.109569472
4	YAL026C	1.042465761	1.021012126	0.895025071	0.752623374	1.042465761	0.678302164
5	YAL027W	1.079228237	0.858565436	1.071773463	0.795536484	1.086734863	0.901250463
6	YAL028W	0.817902059	0.737134609	1.064370182	1.214194884	1.257013375	1.494849249
7	YAL029C	1.028113827	0.846745312	0.864537231	0.852634892	0.920187651	0.806641759
8	YAL030W	0.91383145	0.801069878	1.021012126	0.965936329	1.086734863	1.079228237
9	YAL031C	0.920187651	1.624504793	0.707106781	0.763129604	0.901250463	0.752623374
10	YAL032C	0.852634892	1.042465761	0.920187651	0.972654947	0.835087919	0.920187651
11	YAL033W	0.768437591	1.022173667	0.852634892	0.888842681	0.806641759	0.895025071
12	YAL034C	0.687770909	2.056227653	1.035264924	0.806641759	0.829319546	0.829319546
13	YAL035C-A	1.065049317	1.022173667	1.051613421	1.022549092	1.037123234	1.02211285
14	YAL035W	0.784584098	0.532185091	0.659753955	0.801069878	0.858565436	1.125058485
15	YAL036C	1.028113827	0.582366793	0.757858283	0.659753955	0.933032992	0.784584098
16	YAL037W	1.049716684	0.846745312	1.132883885	1.057018041	1.205807828	1.117287138
17	YAL038W	0.952637998	0.678302164	0.864537231	0.882702996	0.933032992	0.757858283
18	YAL039C	1.283425898	0.986232704	1.140763716	0.732042848	1.042465761	1.172834949
19	YAL001C	0.901250463	0.901250463	0.864537231	1.125058485	0.747424624	0.737134609
20	YAL002W	0.926588062	1.071773463	1.00695555	1.042465761	1.028113827	0.835087919
21	YAL003W	0.907519155	0.611320139	1.071773463	0.801069878	0.757858283	0.668963777
22	YAL004W	0.986232704	0.716977624	0.926588062	1.086734863	0.979420298	1.140763716
23	YAL005C	0.965936329	0.692554734	0.721964598	0.959264119	1.079228237	0.952637998

Figure 3: Screenshot of the dataset values

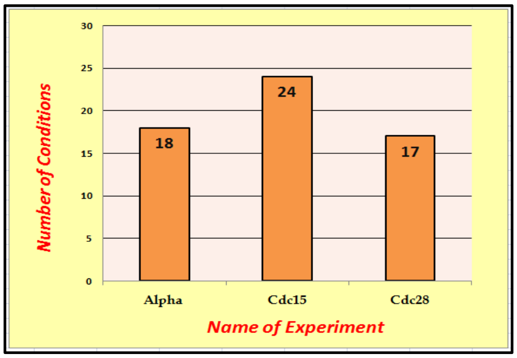


Figure 4: Illustrates the number of Conditions in each Experiment

5.2 Gene Selection

In general, thousands of genes are generated during microarray experiments. Thus, numerous genes are considered to be irrelevant, not useful and sometimes cause certain problems in future analysis processes. Effective gene selection can significantly reduce computational tasks for subsequent processes such as clustering or classification[18]. Due to the huge dimension of microarray data, the dimensionality reduction of this data is a crucial step[9]. In this study, Mutual Information has been used to select informative genes.

5.2.1 Mutual Information Concept

Mutual Information in information theory is defined as a measure of shared dependency between two random variables, which is one of the most effective Genes Selection methods. It determines the amount of information for a given random variable based on the other random variable. The mutual information concept is derived from that of entropy of a random variable[19][20].

Let A be the random variable that has N1 values and B has N2 values. First, the calculation of the entropy of A is given by the entropy H(A,) and the calculation of the entropy of B is given by the entropy H(B), both defined as

$$H(A) = -\sum_{a\in A} P(A)\, log\, p(A) \quad (1)$$

$$H(B) = -\sum_{b\in B} P(B)\, log\, p(B) \quad (2)$$

P(A) are the probabilities values of gene A, and P(B) are the probabilities values of gene B. Second, the calculation of the joint entropy H(A,B), which is the amount of uncertainty associated between two random variables A and B, is defined as follows:

$$H(A,B) = -\sum_{a\in A} \sum_{b\in B} \boldsymbol{P(A,B)\, log\, (A,B)} \qquad (3)$$

P (A, B) represents the joint probability of A and B values occurring together. Finally, The Mutual Information MI (A,B) between the random variables A and B can be calculated as follows:

$$MI\ (A, B) = H(A) + H(B) - H(A, B) \qquad (4)$$

5.3. Clustering Algorithms

Gene expression data clustering is an important objective for biologists and researchers[21]. In this section, some clustering algorithms that have been used in this study are discussed. Informative genes data is selected through the gene selection algorithm (Mutual information). The selected data then are clustered using clustering algorithms.

5.3.1 K-Means Clustering Algorithm

The essential objective of the clustering process is to cluster similar data points into one cluster, assigning the different data points in different groups. It is one of the most common clustering methods. It is a partitioning method used in several applications [22]. It works to assign the objects to a pre-defined number of clusters. The algorithm selects random centers for clusters, one for each. It mostly uses the Euclidean distance to compute the distance between points of data and clusters centroids [23].The computed distance between two data points according to Euclidean distance Q = (q1, q2, q3, …,qn) and P = (p1, p2, p3,...,pn) is described as follows:

$$Dis(Q, P) = \sqrt{\sum_{i=1}^{n}(qi - pi)^2} \qquad (5)$$

```
Algorithm 1: K-Means clustering algorithm.
Input:   D data points, k clusters number.
Output:  K clusters.
Method: Select k objects as the initial centers from D.
Repeat:
    (1) Each data point is assigned to cluster depending on the mean
        value of the data points in the cluster.
    (2) Update the cluster means by calculating the mean value of the
        data points for each cluster.
    (3) Until condition (No change or Maximum iterations) is met.
```

5.3.2 DBSCAN Clustering Algorithm

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is one of the famous clustering techniques based on density to cluster and find the noise in data. One of the advantages of this algorithm is that it does not require the determining of the cluster numbers at the beginning of the algorithm [24]. It depends on two predefined parameter values, whereas MinPts is the lowest objects number in any cluster. Eps is the maximum distance between a particular object and another within the same cluster.

The two parameters are significant for guiding the algorithm and determining the quality of the groups. These two parameters are generally used throughout the whole algorithm, which means that the parameter values are constant for all clusters. The DBSCAN algorithm meets every point in the data [25].

```
Algorithm 2: DBSCAN clustering algorithm
Input:    D objects to be clustered, Eps value, MinPis value.
Output: A set of a cluster of objects.
Method:
    (1) Select a point P.
    (2) Based on Eps and MinPts values retrieve all
        points' density-reachable from P.
    (3) If P is the core point, a cluster is formed
    (4) If p is not the core point, no points are density-
        reachable from P DBSCAN.
         In this case, you will visit the next data point
        within the dataset.
    (5) Continue until all data points have been processed.
```

5.3.3 Mean Shift Clustering

It is a nonparametric clustering technique which does not need predefinition of the cluster numbers, as it is based on kernel density estimation. Generally, the algorithm uses a Gaussian kernel for probability estimation. The algorithm finds out the peaks of the probability distribution. In contrast to other clustering algorithms, the results of mean shift does not depend on any assumptions of points shape distribution, cluster numbers, or random initialization of data points[26].

```
Algorithm 3: Mean shift clustering algorithm
Input:   D a set of data points.
Output:  A set of clusters data.
Repeat:
Define a window and place the window on a data point.
Within the window, the mean of all points is computed.
The window is shifted to the location of the mean.
Repeat step 2-3 until convergence
```

5.4. Clustering Validation

Clustering is an unsupervised approach.The process of evaluating the results of the clustering algorithms is more difficult than the supervised approach itself, as it has the same significance as Clustering algorithms.Clustering validation can provide some indicators to estimate the number of clusters, which represent essential information for cluster analysis. Some clustering validation methods that have been used in this study are reviewed in this section [6].

5.4 .1 Elbow Plot

The Elbow method is a popular method of consistency validation within clusters. The elbow helps to find the optimal clusters number in the data set. This method assumes runningthe clustering algorithm for a range of k values so that the elbow plot can be estimated[27].

```
Elbow Plot Steps
    (1) Run the K-means cluster algorithm for a varied
        range of k-values.
    (2) Compute the Sum of Square Error (SSE) for each
        k clusters and plot curve according to the SSE (K)
        = $\sum_{i=1}^{k}\sum_{p}^{Ci}(p - mi)^2$ equation .where p is data
        point in cluster; mi is the center of $C_i$.
    (3) Until condition (No change or Maximum
        iterations) is met.
```

5.4.2 Silhouette Coefficient

The Silhouette coefficient is the second index that has been used in this study in order to evaluate the results of clustering algorithms. It is a composite index that reflects the cohesion and separation of the clusters[28]. It can apply several distance measures. The Silhouette coefficient for each gene(i) is defined as:

$$S\ (i) = \frac{A(i) - B(i)}{Max\ \{A(i), B(i)\}} \quad (6)$$

Where A(i) represents the mean distance of genes i to other genes within the same group, and B(i) is the mean distance of genes i to genes in the nearest neighbor group.

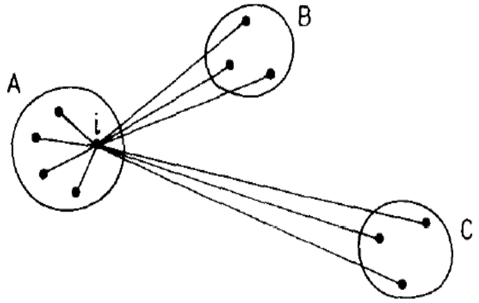


Figure 5: An illustration of the computation of s(i), where the gene i belongs to cluster A[28].

The Silhouette coefficient score can range between -1 (which implies that the clustering is incorrect) and +1 (which indicates that the data points are in an appropriate cluster). The scores close to 0 indicate nested clusters. When the result is high, this indicates that the clusters are dense and well separated[29].

Silhouette Coefficient Steps

(1) Run K-means cluster algorithm for a different range of k values.

(2) For each gene data i, compute the average distance to other genes data in the same cluster A(i).

(3) Compute the average distance of gene i to genes in all other clusters B(i).

(4) Compute he Silhouette coefficient for gene i according to the equation in Section 4.4.2.

(5) Compute the Silhouette coefficient average for all genes data.

VI. Methodology

The proposed approach is divided into four stages: data preprocessing, genes selection, clustering genes, and clustering validation, as shown in Figure 6.

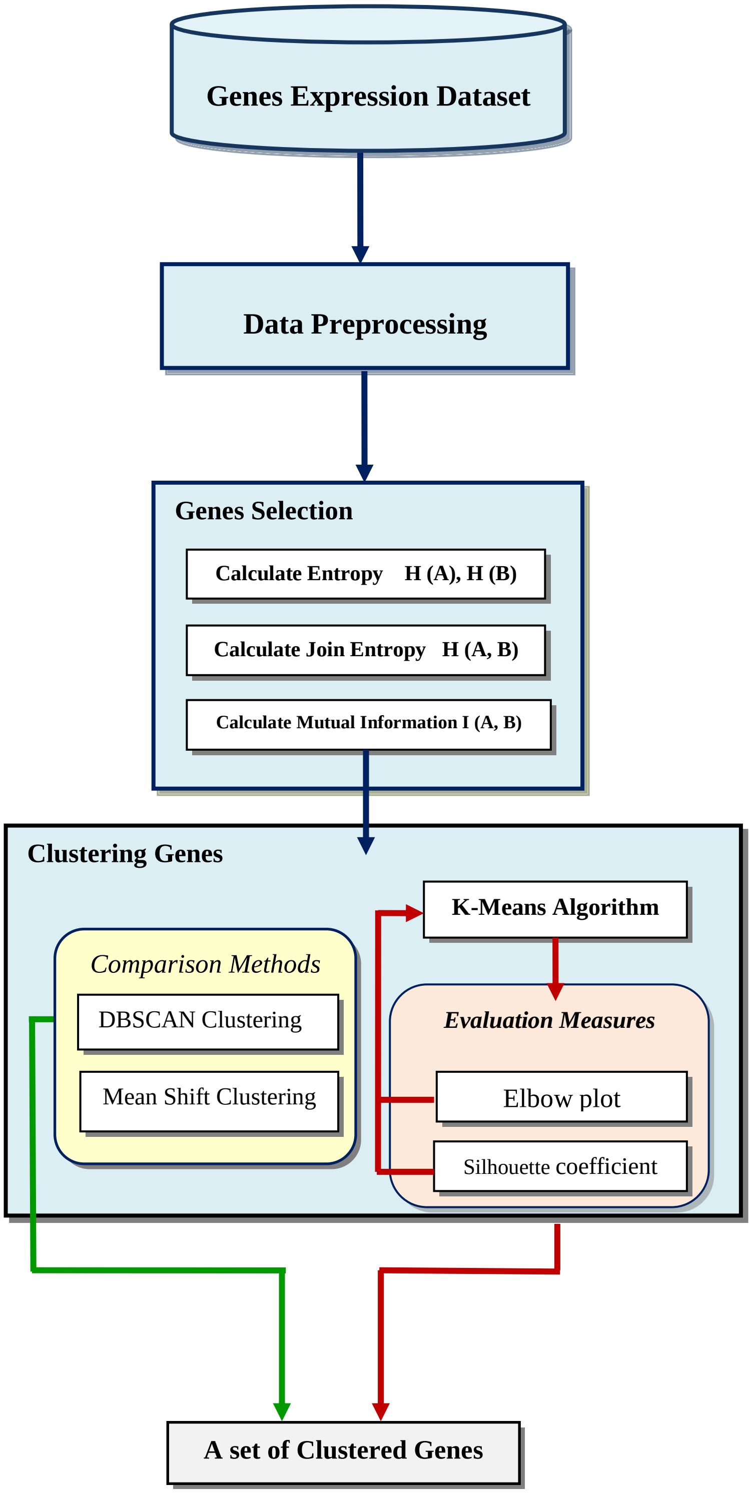


Figure 6: proposed approach

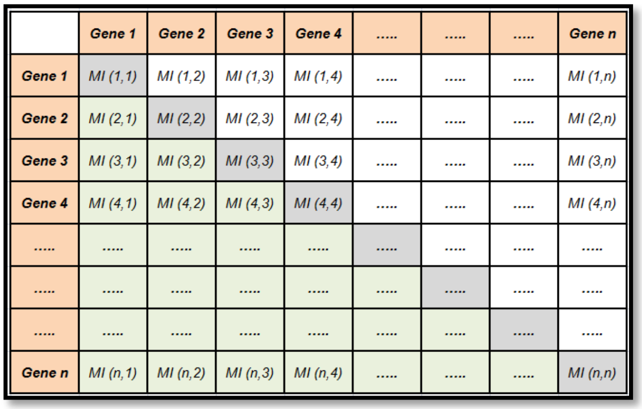
Stage1: Data Pre-processing Huge amounts of data are generated through microarray technology. This data contains a large number of missing values due to several cases such as irregular spots, dust, scratches in the image, and low intensity. In addition, microarray data suffers from noise and outliers values due to different experiences.

In this study, the missing values of gene expression data have been processed by replacing them with the column average according to equation 7.

$$Mean\ (\mu)=\frac{V}{M} \qquad (7)$$

Where V: Sum values in the Column, M:number of items in the column

Stage2: Gene Selection: In this stage, the mutual information has been applied to select the significant genes and to eliminate random genes data, according to the equations in the section. The Mutual Information between all genes is calculated in a dataset. The method involves the build of a two-dimensional matrix n x n where n represents the number of genes in the dataset, as shown in Figure 7.



	Gene 1	Gene 2	Gene 3	Gene 4	.....	.....	.....	Gene n
Gene 1	MI (1,1)	MI (1,2)	MI (1,3)	MI (1,4)	.....	.....	.....	MI (1,n)
Gene 2	MI (2,1)	MI (2,2)	MI (2,3)	MI (2,4)	.....	.....	.....	MI (2,n)
Gene 3	MI (3,1)	MI (3,2)	MI (3,3)	MI (3,4)	.....	.....	.....	MI (3,n)
Gene 4	MI (4,1)	MI (4,2)	MI (4,3)	MI (4,4)	.....	.....	.....	MI (4,n)
.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....
Gene n	MI (n,1)	MI (n,2)	MI (n,3)	MI (n,4)	.....	.....	.....	MI (n,n)

Figure 7: Format of MI matrix

The values of the main diagonal of the matrix are substituted with zero values. The upper triangle values represent the computation of the MI between the genes.The values of the lower triangle represent the same values as those of the upper triangle, thus they are replaced by zero values as well. The mutual information values (upper triangle values) are arranged, and high-ranked genes are selected as input for clustering algorithms[9].

Algorithm 4: Gene Selection

Input: Dn*marray of genes expression, n number of genes,m number of samples, num_genes number of genes required

Output: gen_lis list of genes indexes that have the highest of mutual information. values.

```
Begin
(1)  Set count, X and Y to zero.
(2) Set gen_listo Ø.
(3) Create array of W_MI[n][n]
(4) fori=1 to n //where n: number of genes.
(5) for j=1 to n //where n: number of genes.
(6) if(i< j) then
(7)  for k =1 to m//where m: number of samples.
(8)$G_1[k]$ = D[i][k]//  Store the $gene_1$ vector //
(9)$G_2[k]$ = D[j][k] // Store the $gene_2$ vector //
(10)end for
(11)W_MI[i][j]  =Compute the mutual information between  G1 and G2 according tothe equation (4) in section (4-2-1).
(12) count=count+1  //Number of elements in the upper triangle W_MI
(13)  end if
(14)      end for
(15) end for
//  Store the MI values and  index I and J //
(16)Create array ofMut[count][3]
(17) fori=1 to n
(18)     for j=1 to n
(19)          if (i< j)Then
(20)                 Mut[x][y] = W_MI[i][j]
(21)                 Y=Y+1
(22)                  Mut[x][y] =i
(23)                 Y=Y+1
(24)                  Mut[x][y] =j
(25)                 X=X+1
(26)                 Y=0
(27) end if
(28)      end for
(29)  end for
   //Sort the Mutual Information array Mut[count ][3] in descending order  based on the Mutual Information values//
(30)  Mut = the result of sorting Mut[count ][3]
Repeat(31)
(32)i = i+1
(33)      for j =2 to 3
(34) if (Mut [i][j]Not in gen_lis)Then
(35)             gen_lis=  Mut[i][j]// Add the non-duplicate element to the list//
(36)           end if
(37)     end for
(38)Until (number of elements(gen_lis)<>num_genes)
(39)Returngen_lis
End
```

Stage3: Genes Clustering Algorithms: At this stage, several algorithms of clustering have been implemented to cluster the data of gene expression. The clustering algorithm results are discussed in Section 7.

Stage4: Clustering Validation: The Elbow and Silhouette coefficient methods assume running a clustering algorithm for a range of k values. After that, the optimal number of clusters can be estimated. The cluster validation results are discussed in Section 7.

VII. Result and Discussion

The proposed approach is conducted to demonstrate the effectiveness using mutual information in selecting the informative genes, examining the behavior of different clustering algorithms to cluster gene expression data, and finding the optimal number of clusters.

In the first step, the data of gene expression are read. Then the missing values are processed using the column average. The Mutual Information is calculated between all genes in a yeast cell cycle dataset.

The first 4880 out of 6100 genes selected have high values of MI and are passed to the clustering algorithms as input in the next step. The summary of the reduced data set is presented in Table 2.

Table 2: The summary of the reduced data

Dataset	*Number of Genes*		*Samples*
	original	*reduced*	
Yeast Cell Cycle	*6100*	*4880*	*59*

The k-mean algorithm has been used to divide the data obtained from the previous phase. It is known that the algorithm requires a pre-definition for the number of clusters before running the algorithm.

Elbow Plot and Silhouette Coefficient methods are used to determine the optimal number, where the k-mean algorithm is run more than once.

A range of 2 to 100 of k was used for both the elbow and silhouette plot in this study. The Elbow plot method shows that the maximum change rate of sum of squared error occurs when k = 2 to 8, and stabilizes when the k = 17 to 23.

It shows that the optimal K of data takes place when k = 20.The graph of the Elbow plot for k-Means clustering algorithm is shown in Figure8.Table 3 shows the values of the elbow plot.

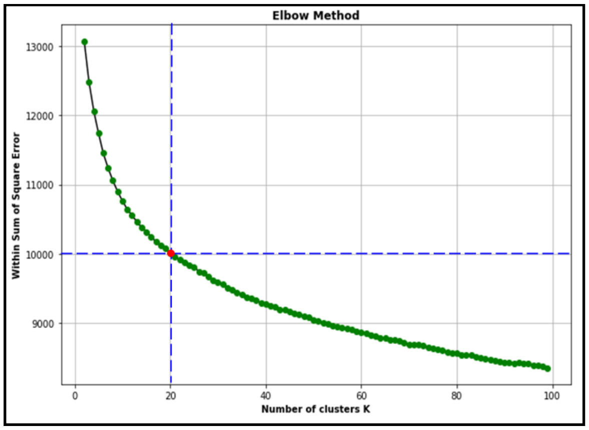


Figure 8: Elbow plot of k-Means clustering algorithm when k=2 to 100

Table 3: Some values of Elbow plot

No. of Clusters	*Sum of Square Error*	*No. of Clusters*	*Sum of Square Error*
K=2	*13073.92*	*K=17*	*10177.25*
K=3	*12486.06*	*K=18*	*10121.95*
K=4	*12064.56*	*K=19*	*10075.27*
K=5	*11740.01*	***K=20***	***10009.43***
K=6	*11450.70*	*K=21*	*9961.41*
K=7	*11237.24*	*K=22*	*9913.41*
K=8	*11055.23*	*K=23*	*9874.44*

The Silhouette coefficient method demonstrates that the highest score is with k=2 clusters and a score around ~ 0.10. It also shows the next highest score when k=3 and a score around ~ 0.07.The graph of the Silhouette Coefficient of the k-Means algorithm is shown in Figure 9.Table 4 shows some values of the Silhouette Coefficient.

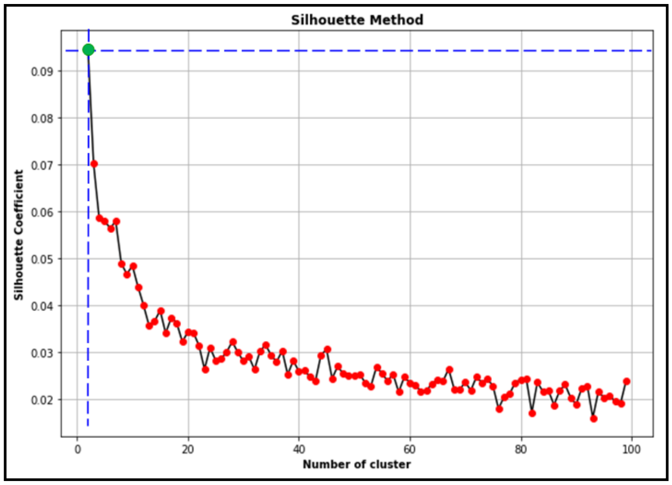


Figure 9: Silhouette Coefficient for k-Means algorithm when k=2 to 100

Table 4: Some values of Silhouette Coefficient

No. of Clusters	*Silhouette Coefficient*	*No. of Clusters*	*Silhouette Coefficient*
K=2	***0.09457***	*K=13*	*0.03565*
K=3	*0.07021*	*K=20*	*0.03427*
K=4	*0.05862*	*K=21*	*0.03416*
K=5	*0.05788*	*K=22*	*0.03140*
K=10	*0.04837*	*K=23*	*0.02637*
K=11	*0.04396*	*K=24*	*0.03094*
K=12	*0.04001*	*K=25*	*0.02825*

The algorithm of k-Means has been applied with k=20 depending on the Elbow plot scores. The algorithm partitions data of genes into 20 clusters. Table 5 shows the results of the algorithm implementation as well as the number of genes for each cluster.Figure10 shows the bar charts of the number of genes for each cluster for k-Mean clustering algorithm.

Table 5: Summary of the results of the K-Mean clustering algorithm

Number of Clusters				***20***	
Cluster 1	**168**	*Cluster 8*	**288**	*Cluster 15*	**272**
Cluster 2	**233**	*Cluster 9*	**221**	*Cluster 16*	**271**
Cluster 3	**396**	*Cluster 10*	**312**	*Cluster 17*	**30**
Cluster 4	**215**	*Cluster 11*	**286**	*Cluster 18*	**352**
Cluster 5	**287**	*Cluster 12*	**197**	*Cluster 19*	**152**
Cluster 6	**234**	*Cluster 13*	**38**	*Cluster 20*	**371**
Cluster 7	**237**	*Cluster 14*	**321**		

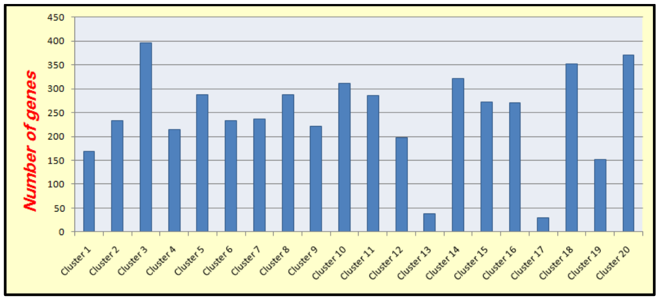


Figure 10: The number of genes in each cluster for k-Mean clustering algorithm

In this work, other algorithms have been implemented to the same reduced data such as Mean shift and DBSCAN algorithms to identify the best partition for microarray data. The algorithm of Mean shift clustering is implemented, which does not require prior knowledge of the number of clusters. The algorithm shows that the number that has been estimated is 13 clusters. Table 6 depicts the results of the Mean shift clustering algorithm implementation. Figure 11 shows the bar charts of the number of genes for each cluster for Mean shift clustering algorithm.

Table 6: Summary of the results of the Mean shift clustering algorithm

Estimate Number of Clusters		*13*	
Cluster1	**4864**	Cluster8	**1**
Cluster2	***1***	*Cluster9*	***1***
Cluster3	***2***	*Cluster10*	***1***
Cluster4	***3***	*Cluster11*	***2***
Cluster5	***2***	*Cluster12*	***1***
Cluster6	***1***	*Cluster13*	***1***
Cluster7	***1***		

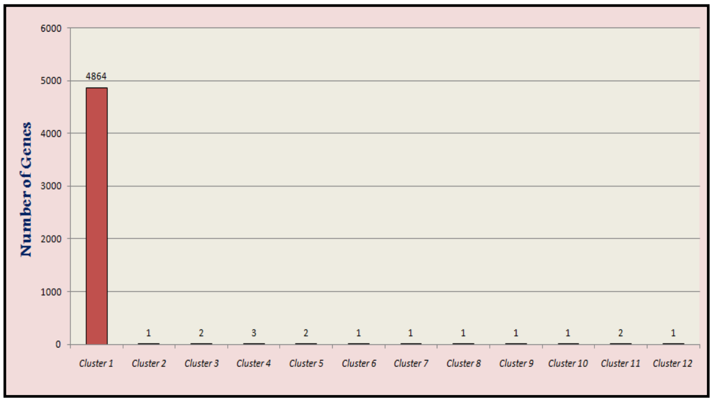


Figure 11: The number of genes in each cluster for Mean shift clustering algorithm

The DBSCAN algorithm does not need any prior knowledge about the number of clusters. The algorithm is dividing the data into2 clusters and 14 noise points. Table-7 illustrates the results of DBSCAN clustering algorithm implementation.Figure12 shows the number of genes for each cluster for DBSCAN clustering algorithm.

Table 7: Summary of the results of DBSCAN clustering algorithm

Estimate Number of Clusters	***1***
Cluster1	***4869***
Noise Points	***12***

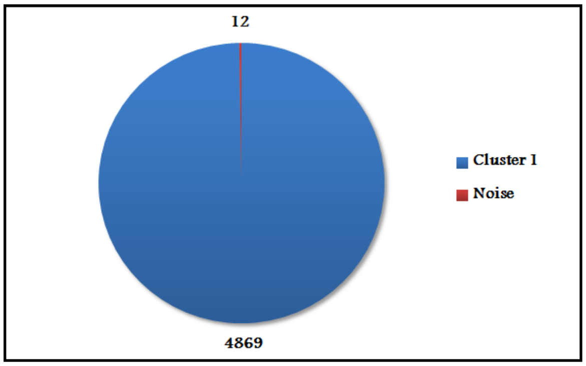


Figure 12: The number of genes in each cluster for DBSCAN clustering algorithm

The proposed approach has been implemented in PYTHON 3.7 and executed in a PC with Intel Core i7 processor with 2.40 GHz speed and 6 GB of RAM.

VIII. CONCLUSION

This study uses the mutual information method of gene selection to obtain the minimum number of random gene and reduce the computational space for improving the quality of the clusters. Then, the K-Means clustering algorithm has been implemented to cluster gene expression data. The validation of clusters is analyzed using Elbow Plot and Silhouette Coefficient. The optimal number of clusters has been selected based on the scores of Elbow plot method.

In comparison with DBSCAN algorithm and the Mean shift algorithm, the results of these algorithms are not quite promising. The data have been clustered in unbalanced clusters to the extent that some of the clusters contained all the data of genes whereas the other clusters contained the least number of genes. The reason is that these algorithms are based on data density in the clustering process. The experimental results show that the proposed approach can eliminate the noises or irrelevant genes data, and it has effectively improved the clustering process.

IX. REFERENCES

[1] D. M. Dziuda, Data mining for genomics and proteomics: analysis of gene and protein expression data, vol. 1. John Wiley & Sons, 2010.

[2] F. Rafii, M. A. Kbir, and B. D. R. Hassani, “Microarray Data Preprocessing To Improve Exploration on Biological Databases,” in International Conference on Big Data, Cloud and Applications, Tetuan, Morocco, 2015, pp. 25–26.

[3] J. Han, J. Pei, and M. Kamber, Data mining: concepts and techniques. Elsevier, 2011.

[4] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," IEEE Trans. Knowl. Data Eng., no. 11, pp. 1370–1386, 2004.

[5] R. Fa, A. K. Nandi, and L.-Y. Gong, "Clustering analysis for gene expression data: A methodological review," in 2012 5th International Symposium on Communications, Control and Signal Processing, 2012, pp. 1–6.

[6] C. Yang, B. Wan, and X. Gao, "Effectivity of internal validation techniques for gene clustering," in International Symposium on Biological and Medical Data Analysis, 2006, pp. 49–59.

[7] S. J. Susmi, H. K. Nehemiah, A. Kannan, and G. Saranya, "Hybrid Algorithm for Clustering Gene Expression Data," Res. J. Appl. Sci. Eng. Technol., vol. 11, no. 7, pp. 692–700, 2015.

[8] T. Scaria, G. Stephen, and J. Mathew, "Gene Expression Data Analysis using Fuzzy C-means Clustering Technique," Int. J. Comput. Appl., vol. 135, no. 8, pp. 33–36, 2016.

[9] A. Makolo and T. Adigun, "Optimization of clustering algorithms for gene expression data analysis using distance measures," Int. J. Comput. Appl., vol. 975, p. 8887, 2016.

[10] J. Parraga-Alava and M. Inostroza-Ponta, "A bi-objective clustering algorithm for gene expression data," CLEI Electron. J., vol. 20, no. 2, pp. 1–17, 2017.

[11] P. Heller and B. Baiju, "An improved distance metric for clustering gene expression time-series data," Am. J. Adv. Res., vol. 2, p. 1, 2018.

[12] N. Yu, Y.-L. Gao, J.-X. Liu, J. Shang, R. Zhu, and L.-Y. Dai, "Co-differential gene selection and clustering based on graph regularized multi-view NMF in cancer genomic data," Genes (Basel)., vol. 9, no. 12, p. 586, 2018.

[13] M. M. Babu, "Introduction to microarray data analysis," Comput. genomics Theory Appl., vol. 17, no. 6, pp. 225–249, 2004.

[14] R. D. Pearson, X. Liu, G. Sanguinetti, M. Milo, N. D. Lawrence, and M. Rattray, "puma: a Bioconductor package for propagating uncertainty in microarray analysis," BMC Bioinformatics, vol. 10, no. 1, p. 211, 2009.

[15] T. Schlitt and P. Kemmeren, "From microarray data to results: Workshop on Genomic Approaches to Microarray Data Analysis," EMBO Rep., vol. 5, no. 5, pp. 459–463, 2004.

[16] Y. Li, W. Liu, Y. Jia, and H. Dong, "A weighted Mutual Information Biclustering algorithm for gene expression data.," Comput. Sci. Inf. Syst., vol. 14, no. 3, pp. 643–660, 2017.

[17] A. Brazma et al., "Minimum information about a microarray experiment (MIAME)—toward standards for microarray data," Nat. Genet., vol. 29, no. 4, p. 365, 2001.

[18] H. Abusamra, "A comparative study of feature selection and classification methods for gene expression data." 2013.

[19] X. Liu, A. Krishnan, and A. Mondry, "An entropy-based gene selection method for cancer classification using microarray data," BMC Bioinformatics, vol. 6, no. 1, p. 76, 2005.

[20] K. Das, J. Ray, and D. Mishra, "Gene selection using information theory and statistical approach," Indian J. Sci. Technol., vol. 8, no. 8, p. 695, 2015.

[21] P. R. Al-Rashid, S., Arifur, M., Al-aaraji, N. H., Lawrence, N. D., & Heath, "Increasing Power by Sharing Information from Genetic Background and Treatment in Clustering of Gene Expression Time Series," J. Univ. Babylon, Pure Appl. Sci., 2018.

[22] C. Zhang and S. Xia, "K-means clustering algorithm with improved initial center," in 2009 Second International Workshop on Knowledge Discovery and Data Mining, 2009, pp. 790–792.

[23] D. Q. Zeebaree, H. Haron, A. M. Abdulazeez, and S. R. M. Zeebaree, "Combination of K-means clustering with Genetic Algorithm: A review," Int. J. Appl. Eng. Res., vol. 12, no. 24, pp. 14238–14245, 2017.

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in Kdd, 1996, vol. 96, no. 34, pp. 226–231.

[25] D. R. Edla, P. K. Jana, and I. S. Member, "A prototype-based modified DBSCAN for gene clustering," Procedia Technol., vol. 6, pp. 485–492, 2012.

[26] K. G. Derpanis, "Mean shift clustering," Lect. Notes, p. 32, 2005.

[27] P. Bholowalia and A. Kumar, "EBK-means: A clustering technique based on elbow method and k-means in WSN," Int. J. Comput. Appl., vol. 105, no. 9, 2014.

[28] T. Thinsungnoena, N. Kaoungkub, P. Durongdumronchaib, K. Kerdprasopb, and N. Kerdprasopb, "The clustering validity with silhouette and sum of squared errors," learning, vol. 3, p. 7, 2015.

[29] R. Lletı, M. C. Ortiz, L. A. Sarabia, and M. S. Sánchez, "Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes," Anal. Chim. Acta, vol. 515, no. 1, pp. 87–100, 2004.

[30] Spellman PT, Sherlock G, Zhang MQ, et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell. 1998;9(12):3273–3297. doi:10.1091/mbc.9.12.3273