International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 29025-29040

http://www.ripublication.com

Hicast-K: A Novel Hybridized Gene Clustering Algorithm With Inbuilt Validation

Muhammad Rukunuddin Ghalib [#1], Motaal Ahmed [*2]
*[# *]School of Computing Science and Engineering*
[#1,2]VIT University, Vellore, Tamil Nadu, India
[1]ghalib.it@gmail.com
[2]motaal.ahmed@gmail.com

Abstract

We went through standard clustering algorithms and selected some conventional algorithms, i.e., k-means method, CAST and hierarchical method, establishing the drawbacks of the mentioned algorithms. We highlighted the computational drawbacks, including optimization, time and space requirements. We then introduce a hybrid clustering algorithm based on the three mentioned clustering methods, with an inbuilt validation. The hybrid algorithm overcomes the major drawbacks of the listed clustering algorithms, and proved to be 25% more efficient than the individual clustering algorithms listed above.

Keyword: Clustering Algorithms, Gene Expression, Microarray, Euclidean Distance, Manhattan Distance, Pearson's Correlation, k-Means, CAST, Hierarchical method, Validation Techniques.

Introduction

A) Data Mining

Data Mining, is widely known as Knowledge Discovery in Databases (KDD). Data Mining and KKD are often mistaken as alternatives, while Data Mining is really just a part of the Knowledge Discovery system. Data mining can be described as a method to analyze patterns, similarities in a given Data Set and to identify and extract useful, unique and original data [1].

In data mining, we provide specific algorithm and plan for the execution of a given data set from a database to discover and extract patterns to analyze primitive and current data and which in turns help to calculate any future developments in the same [2].

B) Bioinformatics

Bioinformatics - a definition (1)

(Molecular) bio – informatics: bioinformatics is conceptualizing biology in terms of molecules (in the sense of physical chemistry) and applying "informatics techniques" (derived from disciplines such as applied math, computer science and statistics) to understand and organize the information associated with these molecules, on a large scale. In short, bioinformatics is a management information system for molecular biology and has many practical applications.

(1)*As submitted to the Oxford English Dictionary*

In today's world, biological data are being generated at an extraordinary rate [3]. With the publishing of H. influenzae genome, entire sequences of over 40 organisms has been unveiled, which ranges up to 100,000 genes [4]. In addition to this, there are countless correlated projects that study gene expressions and protein structure of the genes, the massive data and information being produced is beyond our imagination. Bioinformatics comprises of two terms, Biological Data and Computer Based Calculations.

C) Clustering Techniques

Clustering in data mining is essentially the most important and useful task of the process, which includes obtaining useful patterns and groups for a given data set [5].

One of the major steps during the clustering operation is forming meaningful groups from the obtained patterns and to study the similarities and differences in the given data and to reach a useful and meaningful conclusion [6]. Different sizes of data sets and applications can be tackled using large array of algorithms, proposed in the research [8, 12]. Each type of data set can be grouped effectively with specific algorithms for clustering. However, clustering is an unsupervised process as there is no established example that can validate a desired relation in the outcome [5]. So, the final data sets are constituted relying on the possibilities based on the particular clustering technique.

Considering the expression patterns of the genes, they can be categorized into clusters [16]. There are many established gene clustering methods and also new techniques are being introduced. Some of the gene clustering methods include k-means algorithm, CAST, CLICK, CST, hierarchical clustering, self-organizing map and others. Most of these algorithm faces some drawbacks and prevent obtaining optimal clustering results.

In k-means, the number of clusters, which is nothing but value of 'k', should be known in advance. User has to go through different values of 'k' to reach an optimal number of clusters, which for a very large dataset is not feasible. Secondly, in k-means method, all the elements of a dataset is forced into clusters, which also includes noise and outliers [23, 24]. CAST identifies clusters one after the other, thus finalizing a cluster before moving to the next one. In this process, it focuses on local optima, and fails to satisfy the global optimal value. In hierarchical clustering, one of the most popular method is Eisen's method and is commonly used [7, 25, 26]. But, the agglomerative technique which is widely used fails to deal with errors during

execution [27]. This means that the arrangement of elements in dendrogram heavily relies on the correctness of the dataset and even a small disturbance will widely affect its structure. This also means that once the algorithm starts executing, any undesired selections can't be undone. And lastly, hierarchical clustering suffers from a high time complexity of $O(n^2 \log n)$ [8].

D) Microarray Data Analysis

A group of tiny DNA spots attached to a solid surface is known as microarray. There are thousands of DNA spots in a microarray, involving nearly every gene in a genome.

Various tools have been developed to obtain thousands of genes from a single RNA in the given sample, procedure known as microarray data analysis [9]. A number of steps are involved in the microarray data analysis [10]. Several microarray studies have helped to keep track of expression levels of thousands of genes [11].

Related Work

In this segment, we review the previous associated research with gene clustering. We took a dataset of sixteen genes with five samples of each gene and calculated the clusters using the standard algorithms.

A) K-Means Algorithm

One of the well-known method in data-clustering is K-Means Algorithm. The execution of this algorithm demands the number of clusters to be defined prematurely. This can be considered both as a drawback or convenience. The complete data set can be clustered by specifying the desirable number of clusters beforehand. But to decide the right clustering composition is basically a hit or miss based procedure [12].

Here is a brief description of most common, direct K-Means algorithm also known as Lloyd's algorithm [13]. With initial set of K means $m_1^{(1)},\ldots,m_k^{(1)}$, it involves 2-steps among which the algorithm keeps on shifting [14].

Step 1 known as Expectation step/Assignment Step, the goal is to assign every single data in to clusters, such that the mean of each cluster yields the smallest within-cluster sum of squares, WCSS. Through this, closest mean is automatically achieved as the sum of squares is nothing but the squared Euclidean Distance.

$$S_i^{(t)} = \{x_p : \| x_p - m_i^{(t)} \|^2 \leq \| x_p - m_j^{(t)} \|^2 \ \forall j, 1 \leq j \leq k\}$$

Here, x_p can be designated to more than one $S^{(t)}$ But we assign each one of them to single $S^{(t)}$.

Step 2 known as Maximization Step/Update Step, we find the centroid in the newly formed clusters by calculating mean through the given formula,

$$m_i^{(t+1)} = \frac{1}{| S_i^{(t)} |} \sum_{x_j \in S_i^{(t)}} x_j$$

The new arithmetic mean minimizes the error through the least square method.

Here, the main objective of the algorithm is to enhance or minimize the WCSS goal. The algorithm doesn't necessarily achieve the Global Optima, rather merges at a local optima because of the finite partitioning possibility of the data.

B) K-Medoids Algorithm

A slight variation in K-Means method constitutes the K-Medoids algorithm. In this algorithm, a data element (known as medoid) is selected at each iteration from every cluster.

Quoted [11] "Medoids for each cluster are calculated by finding object *i* within the cluster that minimizes $X_j \in C_i$ *d (i,j)*,

Where C_i is the cluster containing object *i* and *d (i,j)* is the distance between objects *i* and *j*."

A prevalent medoid rule portrays the present clusters as it is. Furthermore, the k-medoids algorithm can look for input distance from a prior formulated distance matrix, thus eliminating the required distance calculations at each iteration [15].

Figures 1 & 2 give the three clusters, the box plot of the genes in the clusters and the expression graph of the genes in the cluster, obtained using this algorithm, respectively.

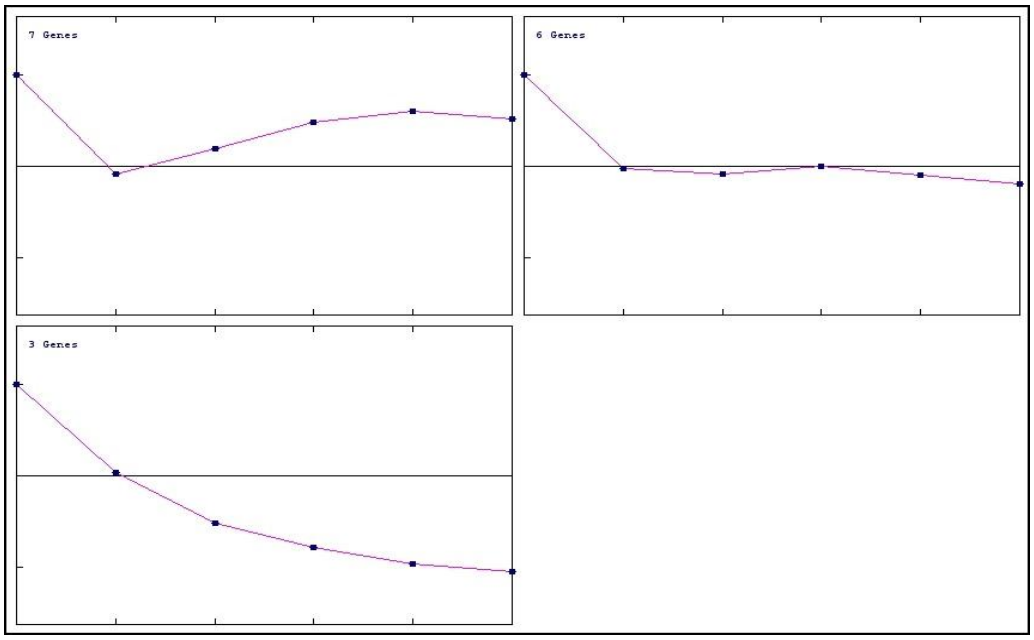

Figure 1: It represents the box plot of the three clusters formed using k-medoids method

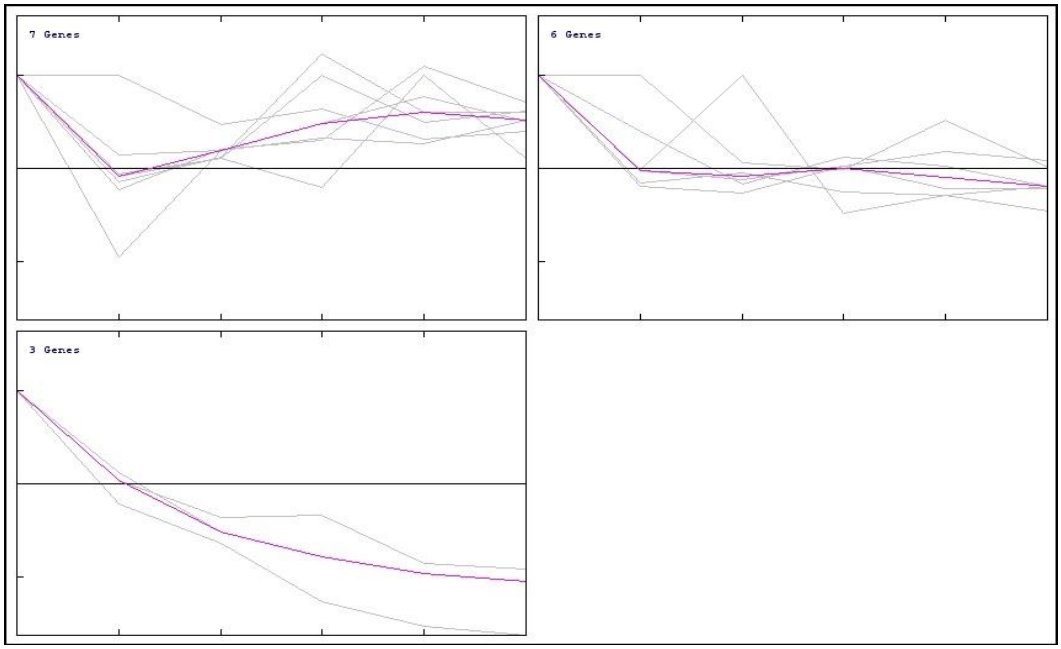

Figure 2: It shows the expression graph of the genes in the three formed clusters

C) Cluster Affinity Search Technique

A constrained clustering technique based on the theoretic concept of *clique* graph data model was proposed by Ben-Dor *et* al[16].

CAST (Cluster Affinity Search Technique) was introduced by Ben-Dor et al [16] as a theoretical procedure and also based on practical heuristics. An *n-by-n similarity matrix* of symmetric nature is provided as an input to the algorithm, and also the affinity threshold *t*. The algorithm selects one cluster at a time, marked as *Copen* and starts searching. The data element ***x***, in the cluster *Copen* is denoted with an affinity value of ***a(x)*** given by:

$$a(x) = \sum_{y \in Copen} S(x, y)$$

If the affinity value of the data element ***x*** equals or is greater than ***t*|*Copen*|**, then it is of high affinity else is of low affinity. Elements having low affinity value ***a(x),*** in ***Copen*** (present cluster), are replaced by high affinity value elements by the algorithm. When the relative low affinity elements have been removed from the present cluster, the process stabilizes and the present cluster is moved to the pool of complete cluster. The algorithm chooses the next cluster and assign it as ***Copen***. When all the data elements are allotted to a cluster, the algorithm stops.

CAST forms complete clusters one after another based on the heuristic search technique and determines the required cluster value based on affinity threshold ***t*** shown in figures 3, 4 & 5. CAST is very efficient in dealing with deviations during the clustering process as it does not rely on the quantity of clusters through user input [17].

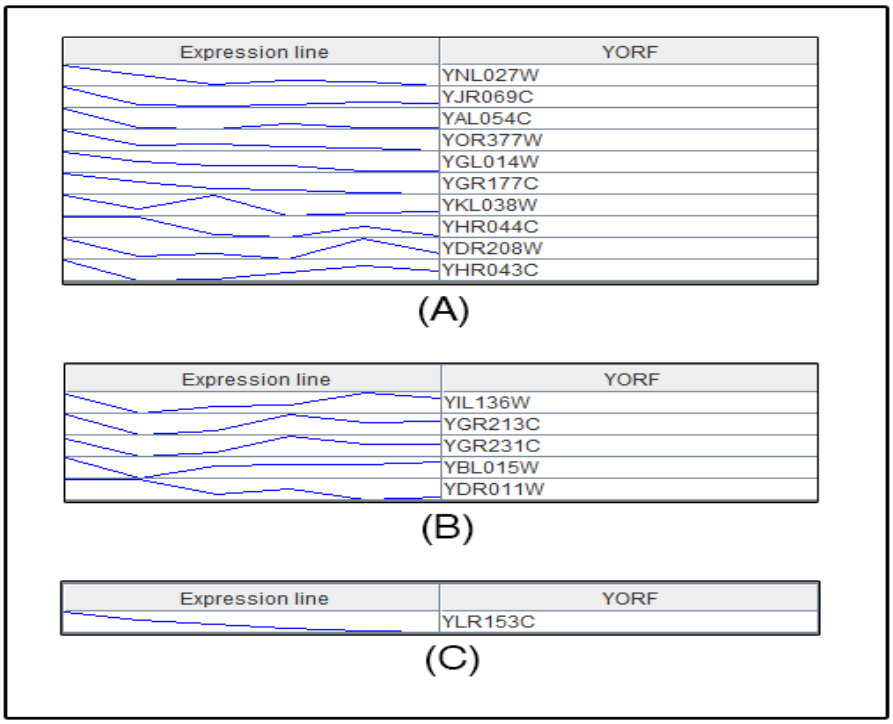

Figure 3: It shows the three formed clusters using CAST

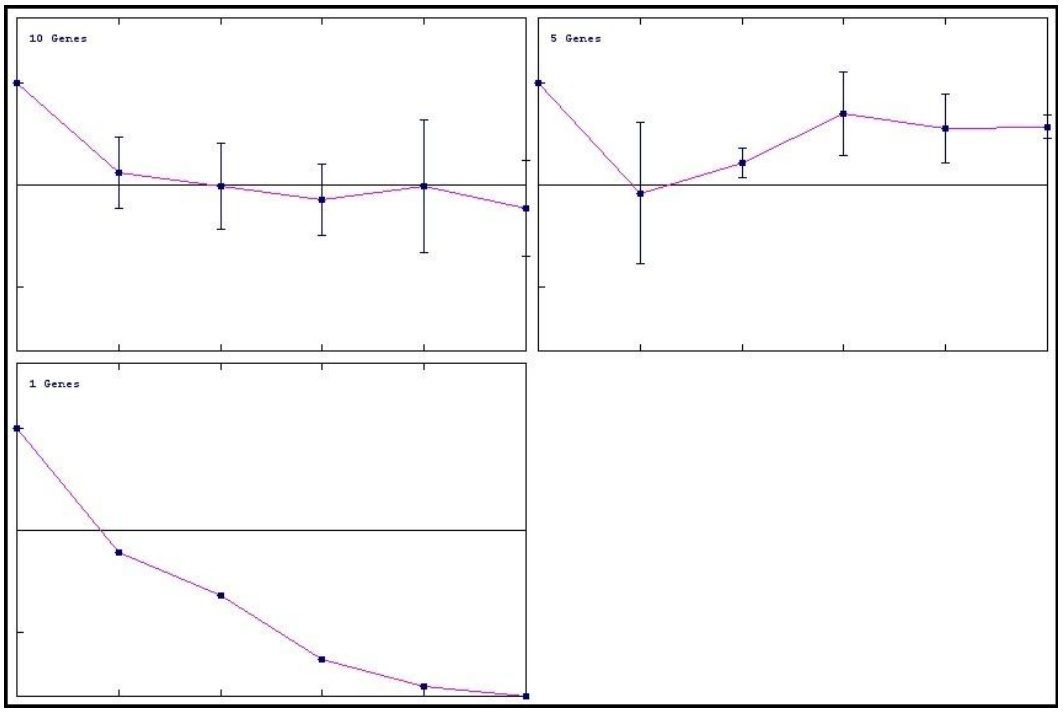

Figure 4: This figure shows the box plot of the genes in the three formed clusters in CAST

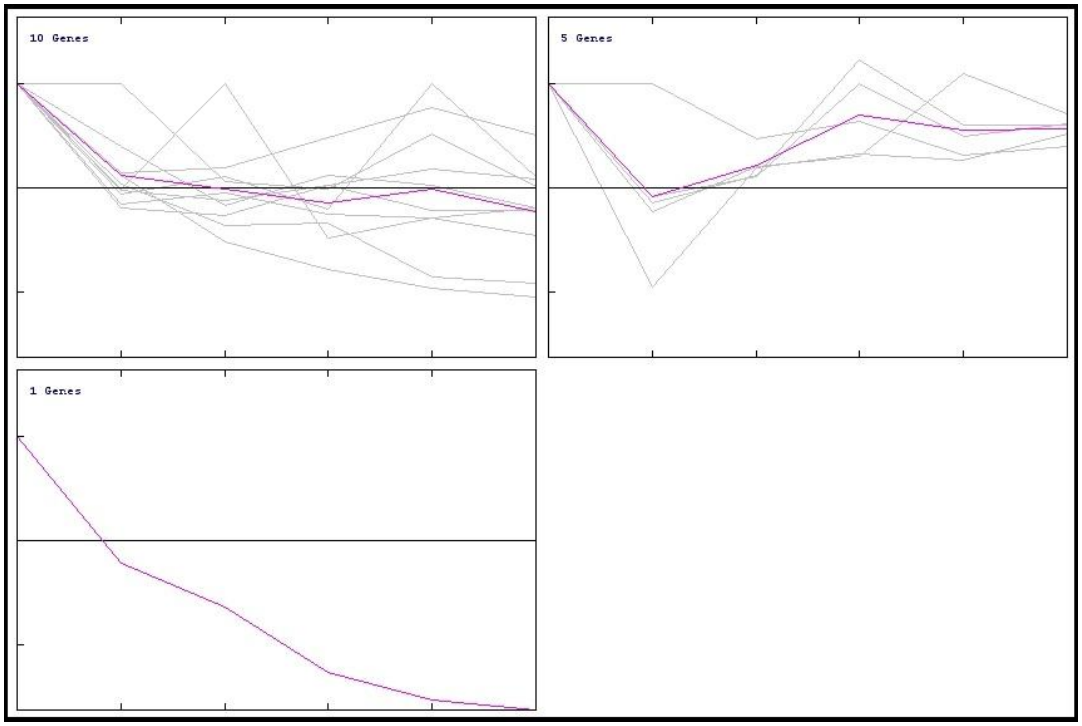

Figure 5: Figure shows the expression graph of the genes in the three clusters in CAST

D) Hierarchical Clustering

Hierarchical Clustering commonly known as Hierarchical Cluster Analysis or HCA, generates hierarchy of clusters shown in figure 6, top comprises of a single complete cluster and sole positioned clusters at the bottom [18].

Hierarchical clustering can be broadly divided in two types:

Agglomerative method is a bottom-up approach, in which the clusters are formed by a chain of merging of objects in the data set. We start with a unique cluster and as we move up the ladder, pairs of clusters are merged together. Second is the divisive method, a top down approach, which forms quality clusters by repeatedly placing a number of elements in them. We begin exploring one cluster and as we travel down the ladder, division of clusters is done repeatedly [19].

Splitting and merging technique is one of the best proposed methods in Hierarchical Cluster Analysis. One of the most favored agglomerative clustering methods is Minimax linkage and as for divisive clustering, average similarity is one of the best technique. For efficient run of the algorithm, an important factor is balanced clusters [20].

Hierarchical Clustering in a Euclidean Space, Quoted [21]

"WHILE it is not time to stop DO

Pick the best two clusters to merge;

Combine those two clusters into one cluster;

END";

To begin, we shall assume the space is Euclidean. That allows us to represent a cluster by its centroid or average of the points in the cluster. Note that in a cluster of one point, that point is the centroid, so we can initialize the clusters straightforwardly. We can then use the merging rule that the distance between any two clusters is the Euclidean distance between their centroids, and we should pick the two clusters at the shortest distance."

We consider two clusters and then choose a pair of genes, one from each cluster. If the genes show the maximum similarity, we consider both the clusters to be similar and this is called Single Link Clustering algorithm. If the resemblance is average among the two, then these are the parameters of cluster similarity in Average Link. And if the resemblance is minimum, then the clusters are grouped together under Complete Link Clustering Algorithm [22].

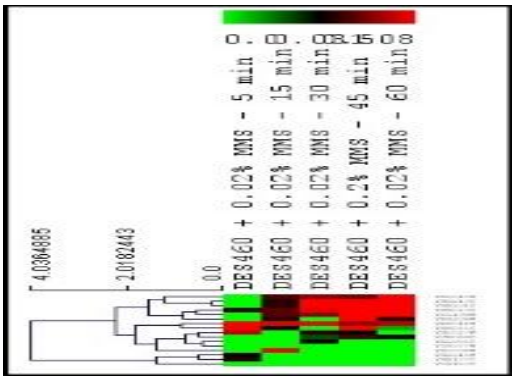

Figure 6: Figure shows the clustering in HCA Tree

Methodology

The Experimental data we used are collected from the website of PNAS web site at (www.pnas.org) or at http://rana.stanford.edu/clustering. We also used the data from Kim lab, Stanford University for our research purpose which is available in http://cmgm.stanford.edu/~kimlab/, (Kim lab). In addition to these datasets, we also have designed synthetic datasets for experimental purpose. The overall architecture of our schematic design is illustrated in figure 7.

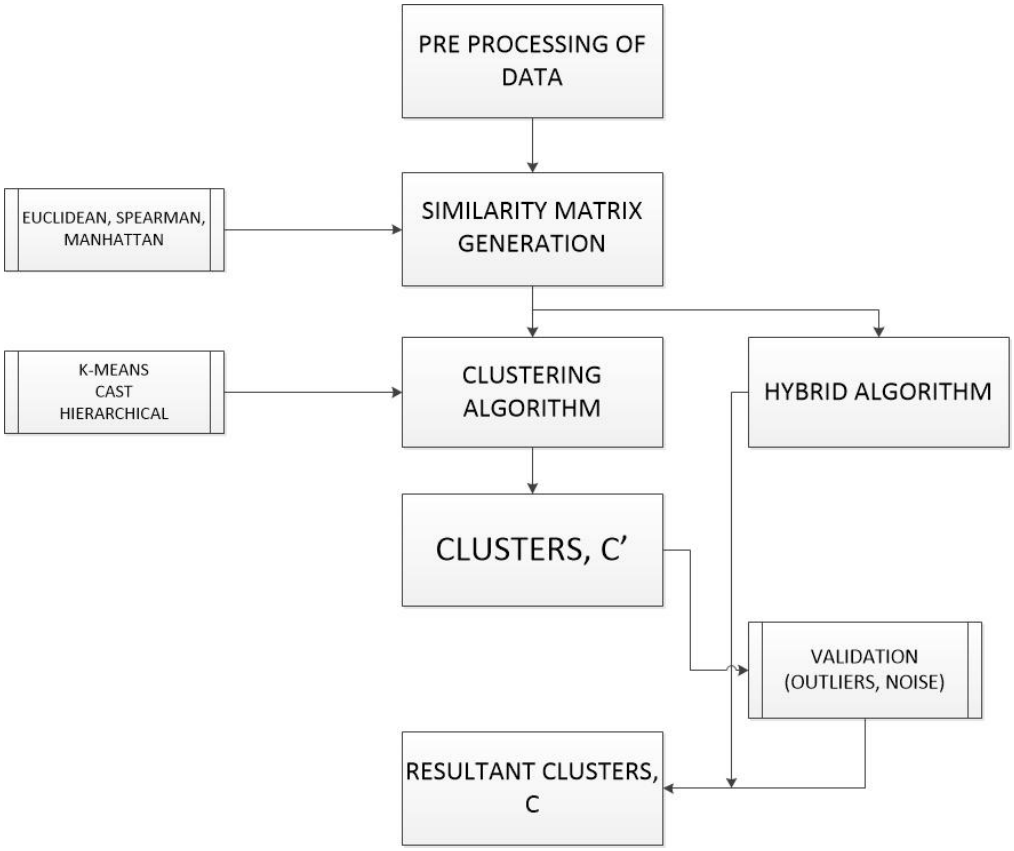

Figure 7: Illustration of the Schematic System Architecture

The similarity matrices were first generated for the dataset, then the matrices have been used to compute the clusters using the standard and hybrid algorithms. Similarity matrix generated using Euclidean distance have been used with all the algorithms in this paper.

Table 1 shows the sample dataset of sixteen genes with five samples each, used in all the experiments performed in this paper.

Table 1: The dataset of 16 genes, each having 5 samples, used in experimented algorithm

YORF	5 min	15min	30 min	45 min	60 min
YNL027W	0.394	-0.17	0.123	0.03	-0.189
YKL038W	-0.01	1	-0.48	-0.284	-0.198
YIL136W	-0.23	0.2	0.302	1.101	0.717
YHR043C	0.15	0.2	0.478	0.766	0.508
YHR044C	0	0.06	-0.01	0.516	0.01
YDR011W	1	0.47	0.64	0.31	0.4

YJR069C	-0.03	-0.12	0.021	0.18	0.088
YGR213C	-0.08	0.1	1	0.498	0.616
YGR177C	0.12	-0.52	-0.779	-0.969	-1.054
YOR377W	-0.154	-0.05	-0.253	-0.286	-0.452
YAL054C	-0.188	-0.27	0.03	-0.22	-0.22
YLR153C	-0.215	-0.64	-1.263	-1.532	-1.633
YBL015W	-0.948	0.19	0.32	0.26	0.52
YDR208W	-0.055	0.11	-0.199	1	0.111
YGR231C	-0.14	0.12	1.227	0.597	0.599
YGL014W	0.032	-0.36	-0.333	-0.854	-0.915

This dataset has been pre-processed using Euclidean distance and Manhattan distance, and corresponding similarity matrices have been generated.

A) Euclidean Distance

If two objects $\vec{O_i}$ and $\vec{O_j}$ are there, then their distance in p-dimensional space is defined as

$$Euclidean(O_i, O_j) = \sqrt{\sum_{d=1}^{P} (O_{id} - O_{jd})^2}$$

Euclidean distance is one of the widely used method to calculate distance between two given objects. The corresponding similarity matrix for our dataset using this formula shown in *table 2*.

Table 2: The similarity matrix for the sample dataset generated using Euclidean distance

	YNL027W	YKL038W	YIL136W	YHR043C	YHR044C	YDR011W	YJR069C	YGR213C	YGR177C	YOR377W	YAL054C	YLR153C	YBL015W	YDR208W	YGR231C	YGL014W
YNL027W	0	0.349	0.393	0.287	0.209	0.299	0.133	0.344	0.411	0.195	0.16	0.657	0.393	0.294	0.394	0.321
YKL038W	0.394	0	0.498	0.442	0.415	0.447	0.332	0.512	0.471	0.275	0.342	0.652	0.428	0.401	0.56	0.407
YIL136W	0.393	0.498	0	0.142	0.389	0.385	0.299	0.234	0.742	0.474	0.423	0.978	0.278	0.202	0.264	0.668
YHR043C	0.287	0.442	0.142	0	0.281	0.252	0.229	0.16	0.68	0.408	0.353	0.917	0.302	0.21	0.205	0.587
YHR044C	0.209	0.415	0.389	0.281	0	0.219	0.272	0.396	0.556	0.372	0.36	0.799	0.51	0.292	0.442	0.492
YDR011W	0.299	0.447	0.385	0.252	0.219	0	0.341	0.304	0.679	0.462	0.428	0.921	0.495	0.392	0.34	0.589
YJR069C	0.133	0.332	0.299	0.229	0.271	0.341	0	0.292	0.46	0.192	0.136	0.693	0.273	0.217	0.346	0.372
YGR213C	0.344	0.512	0.234	0.16	0.396	0.304	0.292	0	0.723	0.437	0.376	0.955	0.281	0.345	0.063	0.615
YGR177C	0.411	0.471	0.752	0.68	0.556	0.679	0.46	0.723	0	0.293	0.356	0.249	0.647	0.606	0.771	0.127
YOR377W	0.195	0.275	0.474	0.408	0.372	0.462	0.192	0.437	0.293	0	0.107	0.514	0.372	0.35	0.501	0.203
YAL054C	0.16	0.342	0.423	0.353	0.36	0.428	0.136	0.376	0.356	0.107	0	0.582	0.318	0.333	0.423	0.256
YLR153C	0.657	0.652	0.978	0.917	0.799	0.921	0.693	0.955	0.249	0.514	0.582	0	0.842	0.827	1	0.248
YBL015W	0.393	0.428	0.278	0.302	0.51	0.495	0.273	0.281	0.647	0.372	0.318	0.842	0	0.331	0.313	0.552
YDR208W	0.294	0.401	0.202	0.21	0.292	0.392	0.217	0.345	0.606	0.35	0.333	0.827	0.331	0	0.387	0.538
YGR231C	0.394	0.56	0.264	0.205	0.442	0.34	0.346	0.063	0.771	0.501	0.423	1	0.313	0.387	0	0.559
YGL014W	0.321	0.407	0.668	0.587	0.492	0.589	0.372	0.615	0.127	0.203	0.256	0.348	0.553	0.538	0.659	0

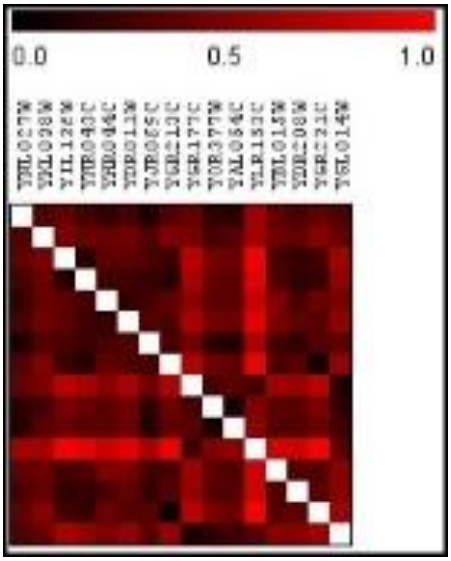

Figure 8: Visualization of Similarity matrix generated using Euclidean distance

B) Manhattan Distance

In Manhattan distance, a grid-like path is followed between two data points and then their distance is measured. It is the submission of differences of all of their respective objects.

$$d = \sum_{j=1}^{m} | X_j - Y_j |$$

Where, m is the total number of variables, X_j and Y_j are the j^{th} element at position X and Y respectively. The corresponding similarity matrix generated using this formula is given in *table 3*.

Table 3: Similarity matrix generated using Manhattan distance for the sample dataset

	YNL027W	YKL038W	YIL136W	YHR043C	YHR044C	YDR011W	YJR069C	YGR213C	YGR177C	YOR377W	YAL054C	YLR153C	YBL015W	YDR208W	YGR231C	YGL014W
YNL027W	0	0.308	0.388	0.295	0.203	0.324	0.123	0.357	0.418	0.2	0.13	0.675	0.35	0.285	0.405	0.323
YKL038W	0.308	0	0.506	0.454	0.423	0.475	0.294	0.499	0.43	0.206	0.252	0.655	0.47	0.347	0.542	0.35
YIL136W	0.388	0.506	0	0.136	0.367	0.363	0.29	0.203	0.74	0.424	0.375	0.913	0.22	0.182	0.212	0.622
YHR043C	0.295	0.454	0.136	0	0.274	0.228	0.241	0.15	0.654	0.406	0.366	0.91	0.219	0.198	0.169	0.559
YHR044C	0.203	0.423	0.367	0.274	0	0.204	0.204	0.34	0.589	0.342	0.311	0.846	0.391	0.231	0.383	0.494
YDR011W	0.324	0.475	0.363	0.228	0.204	0	0.33	0.273	0.743	0.495	0.455	1	0.335	0.399	0.316	0.648
YJR069C	0.123	0.294	0.29	0.241	0.204	0.33	0	0.258	0.449	0.181	0.126	0.669	0.251	0.161	0.306	0.332
YGR213C	0.357	0.499	0.203	0.15	0.34	0.273	0.258	0	0.708	0.411	0.371	0.915	0.243	0.275	0.052	0.591
YGR177C	0.418	0.43	0.74	0.654	0.589	0.743	0.449	0.708	0	0.315	0.364	0.256	0.701	0.557	0.756	0.117
YOR377W	0.2	0.206	0.424	0.406	0.342	0.495	0.181	0.411	0.315	0	0.103	0.504	0.385	0.266	0.444	0.198
YAL054C	0.13	0.252	0.375	0.366	0.311	0.455	0.126	0.371	0.364	0.103	0	0.544	0.337	0.282	0.404	0.247
YLR153C	0.675	0.655	0.913	0.91	0.846	1	0.669	0.915	0.256	0.504	0.544	0	0.875	0.77	0.948	0.351
YBL015W	0.35	0.47	0.22	0.219	0.391	0.335	0.251	0.243	0.701	0.385	0.337	0.875	0	0.326	0.271	0.584
YDR208W	0.285	0.347	0.182	0.198	0.231	0.399	0.161	0.275	0.557	0.266	0.282	0.77	0.326	0	0.698	0.44
YGR231C	0.405	0.542	0.212	0.169	0.383	0.316	0.306	0.052	0.756	0.444	0.404	0.948	0.271	0.298	0	0.639
YGL014W	0.323	0.35	0.622	0.559	0.494	0.648	0.332	0.591	0.117	0.198	0.247	0.351	0.584	0.44	0.639	0

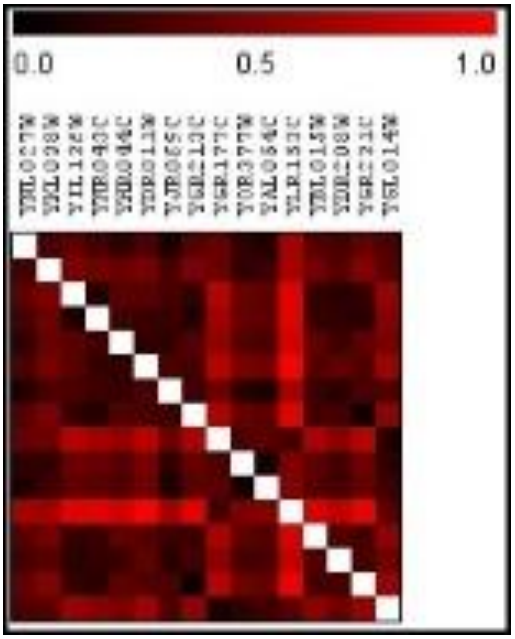

Figure 9: Similarity matrix generated using Manhattan distance

The proposed algorithm is a hybrid algorithm based on k-means, CAST and Hierarchical methods.

Hi Cast-K Algorithm with inbuilt validation

- Randomly select k- cluster centers
- Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.
- Assign each data point to the cluster Ci corresponding to the closest cluster representative (center) ($1 \leq i \leq k$)
- Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to

d[(r),(s)=mind[(i),(j)]

Where the minimum is over all pairs of clusters in the current clustering.

- Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r,s) and old cluster (k) is defined in this way:

d[(k), (r,s)] = min d[(k),(r)], d[(k),(s)]

- while a close gene i not in C or distant gene i in C exists
- Find the nearest close gene i not in C and add it to C
- Remove the farthest distant gene i in C
- Validate C using validation techniques

- Repeat the whole process
- Collect final clusters, **C**, which are validated

Results and Discussion

This algorithm operates parallel to the Euclidean Distance, i.e. allocating data to the nearest clusters based on distance. It is shown in figure 10 & 11

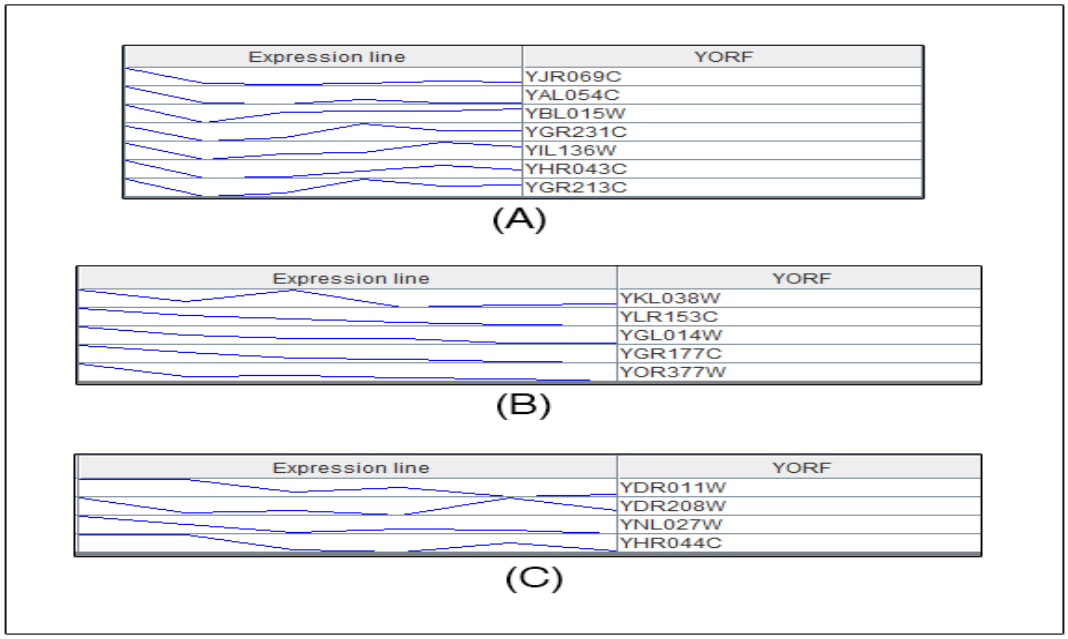

Figure 10: Three formed clusters, A, B and C respectively, using HiCast-K method

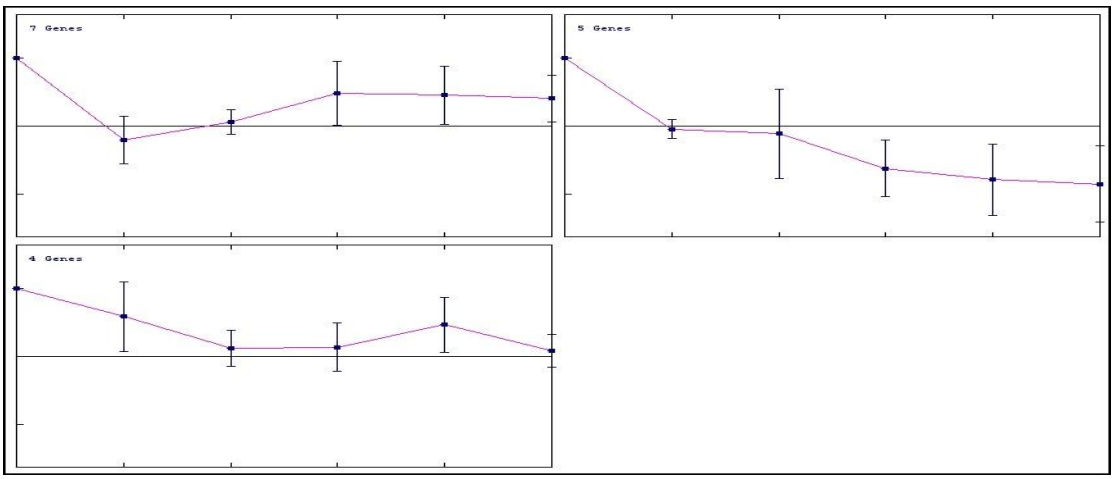

Figure 11: Box Plot of the genes in the three formed clusters using the HiCast-K method

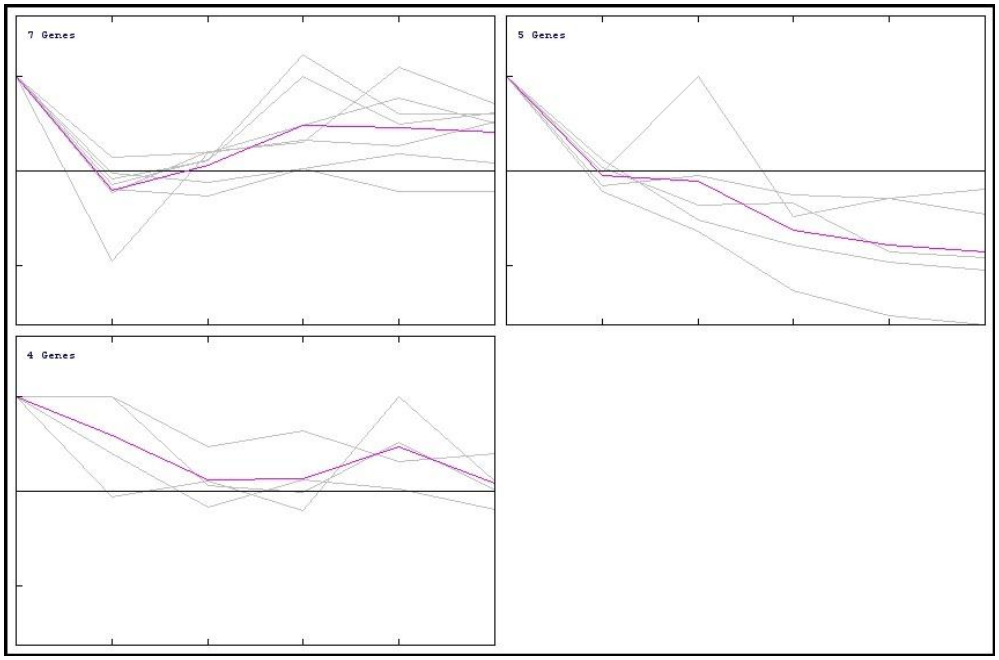

Figure 12: Figure shows the expression graph of the genes in the three clusters

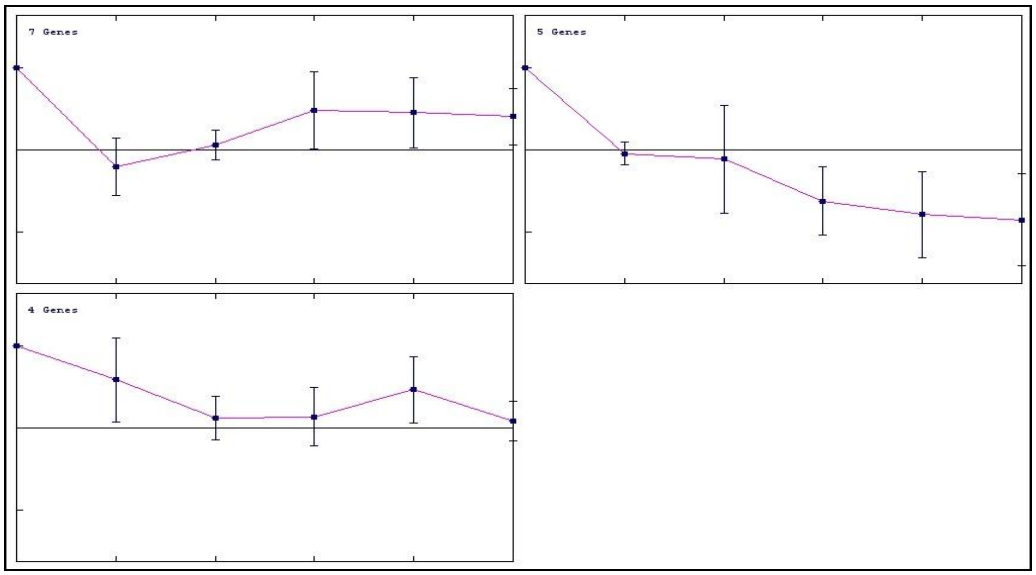

Figure 13: The three clusters A, B and C respectively, formed using HiCast- k-medoids method

Conclusion and Future Work

In this paper, we have proposed a new hybrid clustering algorithm based on standard K-Means, CAST and Hierarchical clustering method. This algorithm tackles the major drawbacks faced by the conventional algorithms. The clustering done using this

hybrid algorithm, is of enhanced quality and an overall improvement of 25% in efficiency is shown over the conventional algorithms.

In future, we could introduce rough and fuzzy concept in our algorithm to tackle the vagueness, missing values, and trimming features issues.

References

[1] Sunil Babu Guntur. Study of Clustering Algorithms for Gene Data Analysis. May 2007.

[2] Y. Ramamohan, K. Vasantharao, C. Kalyana Chakravarti, A.S.K.Ratnam. A study of data mining tools in knowledge discovery process. IJSCE, ISSN: 2231-2307, pp 191-194, July 2012.

[3] Reichhardt T. It's sink or swim as a tidal wave of data approaches. Nature 1999;399(6736):517-20.

[4] Fleischmann RD, Adams MD, White O, Clayton RA, Kirkness EF, Kerlavage AR, et al. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. Science 1995;269 (5223):496-512.

[5] Michad J. A. Berry, Gordon Linoff . Data Mining Techniques For marketin~ Sales and Customer Support. John Willey & Sons, Inc, 1996.

[6] Guha, S, Rastogi, R., Shim IcL. "ROCK: A Robust Clustering Algorithm for Categorical Attributes", Published in the Proceech~gs of the IEEE Conference on Data Engineering, 1999.

[7] Eisen, Michael B., Spellman, Paul T., Brown, Patrick O. and Botstein, David . Cluster analysis and dis- play of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA, 95(25):14863–14868, December 1998

[8] Jain, A.K., Murty, M.N., Flyn, P.J.. "Data Clustering: A Review", ACM Coorputittg Surve2s, Vol.31, No3, 1999.

[9] G.Baskar, Dr.P. Ponmuthuramakingam. A comparative study and analysis for Microarray Gene expression data using Clustering Techniques. IJETTCS. ISSN 2278-6856. Pp: 321-323. Volume 2, June 2013.

[10] P.Venkatesan, Jamal Fathima .J.I. Identification of differentially expressed genes by unsupervised learning method. IJDMTA. ISSN: 2278-2419. Pp. 121-125. Vol 02, June 2013.

[11] Sajid Nagi, Dhruba K. Bhattacharyya, Jugal K. Kalita. Gene Expression Data Clustering Analysis: A Survey.

[12] Han, J. and Kamber, M. Data Mining: Concepts and Techniques, 2000 (Morgan Kaufmann, San Francisco, California).

[13] Lloyd, Stuart P. (1982), *Least squares quantization in PCM*, *IEEE Transactions on Information Theory* **28** (2): 129–137

[14] MacKay, David (2003). "Chapter 20. An Example Inference Task: Clustering". Information Theory, Inference and Learning Algorithms. Cambridge University Press. Pp. 284-292. ISBN 0-521-64298-1.

[15] A. P. Reynolds, G. Richards, and V. J. Rayward-Smith. The Application of K-medoids and PAM to the Clustering of Rules.

[16] Ben-Dor A., Shamir R. and Yakhini Z. Clustering gene expression patterns. Journal of Computational Biology, 6(3/4):281–297, 1999.
[17] Daxin Jiang Chun Tang Aidong Zhang. Cluster Analysis for Gene Expression Data: A Survey. State University of New York, Buffalo.
[18] George Karypis Eui-Hong (Sam) Han Vipin Kumar. A Hierarchical Clustering Algorithm Using Dynamic Modeling. Department of Computer Science and Engineering, University of Minnesota.
[19] Cheng Y., Church GM., 2000, Biclustering of expression data. Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB), 8:93– 103, 2000.
[20] Chris ding and Xiaofeng He (2002), Cluster Merging and Splitting in Hierarchical Clustering Algorithms.
[21] Clustering - The Stanford University InfoLab, http://infolab.stanford.edu/~ullman/mmds/ch7.pdf
[22] Ka Yee Yeung, David R. Haynor, Walter L. Ruzzo. Details of the Clustering Algorithms Supplement to the paper "Validating Clustering in Gene Expression Data" (to appear in Bioinformatics).
[23] Smet, Frank De, Mathys, Janick, Marchal, Kathleen, Thijs, Gert, Moor, Bart De and Moreau, Yves. Adaptive quality-based clustering of gene expression profiles. Bioinformatics, 18:735–746, 2002.
[24] Sherlock G. Analysis of large-scale gene expression data. Curr Opin Immunol, 12(2):201–205, 2000.
[25] Perou C.M., Jeffrey S.S., Rijn, M.V.D., Rees C.A., Eisen M.B., Ross D.T., Pergamenschikov A., Williams C.F., Zhu S.X., Lee, J.C.F., Lashkari D., Shalon D., Brown P.O., and Bostein D. Distinc- tive gene expression patterns in human mammary epithelial cells and breast cancers. Proc. Natl. Acad. Sci. USA, Vol. 96(16):9212–9217, August 1999.
[26] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D. and Levine A.J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonu- cleotide array. Proc. Natl. Acad. Sci. USA, Vol. 96(12):6745–6750, June 1999.
[27] Tamayo P., Solni D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E.S. and Golub T.R Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proc. Natl. Acad. Sci. USA, Vol. 96(6):2907–2912, March 1999.